# Novel Methods for Microscopic Image Processing, Analysis, Classification and Compression

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Alexander Suhre

March 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Ahmet Enis Çetin (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Orhan Arıkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Uğur Güdükbay

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Asst. Prof. Dr. Pınar Duygulu-Şahin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Aydın Alatan

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of Graduate School of Engineering and Science

# ABSTRACT

## Novel Methods for Microscopic Image Processing, Analysis, Classification and Compression

Alexander Suhre
Ph.D. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Ahmet Enis Çetin
March 2013

Microscopic images are frequently used in medicine and molecular biology. Many interesting image processing problems arise after the initial data acquisition step, since image modalities are manifold. In this thesis, we developed several algorithms in order to handle the critical pipeline of microscopic image storage/compression and analysis/classification more efficiently.

The first step in our processing pipeline is image compression. Microscopic images are large in size (e.g. 100K-by-100K pixels), therefore finding efficient ways of compressing such data is necessary for efficient transmission, storage and evaluation.

We propose an image compression scheme that uses the color content of a given image, by applying a block-adaptive color transform. Microscopic images of tissues have a very specific color palette due to the staining process they undergo before data acquisition. The proposed color transform takes advantage of this fact and can be incorporated into widely-used compression algorithms such as JPEG and JPEG 2000 without creating any overhead at the receiver due to its

DPCM-like structure. We obtained peak signal-to-noise ratio gains up to 0.5 dB when comparing our method with standard JPEG.

The next step in our processing pipeline is image analysis. Microscopic image processing techniques can assist in making grading and diagnosis of images reproducible and by providing useful quantitative measures for computer-aided diagnosis. To this end, we developed several novel techniques for efficient feature extraction and classification of microscopic images.

We use region co-difference matrices as inputs for the classifier, which have the main advantage of yielding multiplication-free computationally efficient algorithms. The merit of the co-difference framework for performing some important tasks in signal processing is discussed. We also introduce several methods that estimate underlying probability density functions from data. We use sparsity criteria in the Fourier domain to arrive at efficient estimates. The proposed methods can be used for classification in Bayesian frameworks. We evaluated the performance of our algorithms for two image classification problems: Discriminating between different grades of follicular lymphoma, a medical condition of the lymph system, as well as differentiating several cancer cell lines from each another. Classification accuracies over two large data sets (270 images for follicular lymphoma and 280 images for cancer cell lines) were above 98%.

*Keywords: Image compression, feature extraction, pattern recognition, kernel density estimation, region covariance matrix*

# ÖZET

## MİKROSKOPİK İMGE İŞLEME, ANALİZ, SINIFLANDIRMA VE SIKIŞTIRMA İÇİN YENİ YÖNTEMLER

Alexander Suhre
Elektrik ve Elektronik Mühendisliği Bölümü Doktora
Tez Yöneticisi: Prof. Dr. Ahmet Enis Çetin
Mart 2013

Mikroskopik imgeler tıp biliminde ve moleküler biyoloji alanında sıklıkla kullanılır. İmge özellikleri değişkenlik gösterdiği için de pek çok ilgi çekici imge işleme problemleri ortaya çıkar. Bu tezde, mikroskopik imge depolama/sıkıştırma ve analiz/sınıflandırma sürecini ele almak için bazı algoritmalar geliştirdik.

Sürecimizdeki ilk adım imge sıkıştırmadır. Mikroskopik imgelerin boyutları büyüktür (örneğin 100K'ya 100K piksel). Dolayısıyla da, bu imgelerin verimli bir şekilde iletimi, saklanması ve değerlendirilebilmesi için etkin sıkıştırma yöntemlerinin geliştirilmesi gereklidir.

Verilen bir imgenin renk içeriğini kullanarak blok-uyarlamalı renk dönüşümü uygulayan bir imge sıkıştırma yöntemi öneriyoruz. Veri ediminden önce uygulanan boyama süreci sebebiyle dokuların mikroskopik imgeleri kendine özgü bir renk paletine sahiptir. Önerdiğimiz renk dönüşümü bu özellikten faydalanır ve JPEG ya da JPEG 2000 gibi yaygın olarak kullanılan sıkıştırma algoritmalarına eklenebilir. Bu ekleme alıcıda DPCM'e benzer yapısı yüzünden ek bir yük getirmez. Bu yöntemle, standart JPEG ile kıyaslayınca sinyal-gürültü oranında 0.5 dB'ye varan miktarda artış elde ettik.

Sürecimizdeki sonraki adım imge analizidir. Mikroskopik imge işleme teknikleri imgelerin sınıflandırmasını ve işlenmesini tekrarlanabilir kılar ve bilgisayar destekli teşhiste faydalı sayısal ölçekler sağlar. Bu amaçla, etkili öznitelik çıkarma ve mikroskopik imge sınıflandırılması için çeşitli yeni teknikler geliştirdik.

Burada çarpma gerektirmeyen, hesaplama açısından verimli algoritmalar kullanma avantajını sağlayan, alan tabanlı eş-farklılık matrislerini sınıflandırıcıya girdi olarak verme yöntemini kullanıyoruz. Eş-farklılık matrisleri çerçevesinin, işaret işlemedeki temel bir takım işlerin yapılmasındaki önemini tartışıyoruz. Ayrıca altta yatan olasılık yoğunluk fonksiyonlarını veriden tahmin eden bir takım yöntemler öneriyoruz. Fourier düzlemindeki seyreklik kriterlerini etkili tahminlere ulaşmakta kullanıyoruz. Önerdiğimiz yöntemler Bayes çerçevesinde sınıflandırma için kullanılabilir. Algoritmalarımızın performansını iki imge sınıflandırma probleminde değerlendirdik: Foliküler lenfomanın farklı derecelerini ayırt etmek ve bir takım kanser hücre dizilerini birbirinden ayırmak. İki büyük veri setinde (foliküler lenfoma için 270 imge ve kanser hücre dizileri için 280 imge) sınıflandırma doğruluğu %98'in üzerindedir.

*Anahtar Kelimeler: imge sıkıstırma, öznitelik çıkarımı, örüntü tanıma, çekirdek yoğunluk kestirimi (KDE), alan değişinti matrisi.*

# ACKNOWLEDGMENTS

As John Donne once said: "No man is an island." The following people have been indispensable to the learning process I underwent during my time in graduate school. I would like to thank

- My supervisor A. Enis Çetin for supporting me in every possible way.
- Orhan Arıkan and Uğur Güdükbay for their valuable feedback during the work on this thesis.
- Pınar Duygulu Şahin and Aydın Alatan for agreeing to be on my defense committee.
- Levent Onural for giving me the chance to start graduate school at Bilkent University.
- Ergin Atalar for his help when the going got tough.
- Rengül Çetin-Atalay for giving me the opportunity to work in the field of microscopic images.
- The Scientific and Technological Research Council of Turkey (TÜBİTAK) for my scholarship.
- The European Commission and its IRSES Project 247091 MIRACLE for additional funding.

- My office mates during the first years of my Ph.D., for helping me adjust to the new environment and thereby teaching me many things. I will always be indebted to Gökhan Bora Esmer, Fahri Yaraş, Erdem Ulusoy, Ali Özgür Yöntem and Erdem Şahin.

- Kıvanç Köse and his wonderful wife Bilge for being just the way they are.

- Ayça Özçelikale and Mehmet Köseoğlu for their friendship during hard times and the delicious fruit.

- Mürüvet Parlakay for her tireless efforts and good humor.

- M. Furkan Keskin and Tülin Erşahin for their collaboration on many experiments.

- All members of the Signal Processing Group Bilkent past and present: Behçet Uğur Töreyin, Osman Günay, Serdar Çakır, Onur Yorulmaz, Alican Bozkurt, İhsan İnaç, Y. Hakan Habiboğlu, Ahmet Yazar, R. Akın Sevimli, Fatih Erden, Kaan Duman, Onay Urfalioğlu and Kasım Taşdemir.

- All the good people I met in Ankara: Zeynep Yücel, Elif Aydoğdu, Aslı Ünlügedik, Gülis Zengin, Can Uran, Esra Abacı-Türk & Ata Türk, S. Figen Öktem, Ezgi Şafak, Teksin Acıkgöz Kopanoğlu & Emre Kopanoğlu, Aydan Ercingöz, Sıla Türkü Kural Ünal & Alper Ünal, Gökçe Balkan, M. Can Kerse, Sinan Taşdelen, Yiğitcan Eryaman, A. Serdar Tan, Namık Şengezer, Volkan Hünerli, Merve Önenli Güven & Volkan Güven, Anirudh Srinivasan, Nurdane Öz, Deniz Atasoy, Simon Sender, Kevin Kusmez & Gülsüm Özsoy Kusmez, Natalie Rizzuto and Julie Shaddox for the good times and many laughs.

- John Angliss for reminding me from time to time that there is more.

- Sibel Kocaoğlan for trying all she could to keep me sane.

- D. Robert Iskander for telling me that I could do it.

- Marshall Govindan and Shibendu Lahiri for their advice and insights.
- Christian Hahn for "keepin' it real".
- My parents Leonie & Rüdiger as well as the rest of my family for their love and understanding.

Last but not least, my heart goes out to Yufang Sun. Her love and friendship have sustained me throughout these challenging times.

# Contents

**1 INTRODUCTION** **1**

1.2 Microscopic Image Compression . . . . . 3

1.3 Microscopic Image Analysis . . . . . 4

1.3.1 Feature Extraction . . . . . 5

1.4 Microscopic Image Classification . . . . . 6

1.4.1 Follicular Lymphoma Classification Using Sparsity Smoothing . . . . . 7

1.4.2 Bandwidth Selection in Kernel Density Estimation using Fourier Domain Constraints . . . . . 8

1.4.3 Image Thresholding using Mixtures of Gaussians . . . . . 9

1.4.4 Cancer Cell Line Classification using Co-Difference Operators . . . . . 10

**2 MICROSCOPIC IMAGE COMPRESSION** **13**

2.1 Algorithm . . . . . 16

2.2 Experimental Studies and Results . . . . . 22

**3 CLASSIFICATION OF FOLLICULAR LYMPHOMA IMAGES BY REGION COVARIANCE AND A BAYESIAN CLASSIFIER USING SPARSITY IN TRANSFORM DOMAIN 39**

3.1 Feature Extraction . . . . . 42

3.2 Classification . . . . . 45

3.2.1 First classification stage . . . . . 45

3.2.2 Second classification stage . . . . . 45

3.3 Experimental Results . . . . . 49

**4 BANDWIDTH SELECTION IN KERNEL DENSITY ESTIMATION USING FOURIER DOMAIN CONSTRAINTS 54**

4.1 CV-based cost function for Bandwidth Estimation . . . . . 57

4.1.1 Cross-validation using $l_1$ Norm in Fourier Domain . . . . . 58

4.1.2 Cross-validation using $l_1$ Norm and Filtered variation . . . 59

4.2 Simulation Results . . . . . 60

**5 IMAGE THRESHOLDING USING MIXTURES OF GAUSSIANS 77**

5.1 Description of the Algorithm . . . . . 78

5.1.1 Bandwidth choice of the Gaussian kernel . . . . . 80

5.2 Experimental Results . . . . . 81

**6 CANCER CELL LINE CLASSIFICATION USING CO-DIFFERENCE OPERATORS** **91**

6.1 Vector Product and Image Feature Extraction Algorithm . . . . . 92

6.2 Experimental results . . . . . 95

6.2.1 Feature Extraction . . . . . 97

**7 CONCLUSIONS** **103**

# List of Figures

2.1 A general description of our prediction scheme. To predict the color content of the black-shaded image block, color contents of previously encoded gray-shaded blocks, marked by arrows, are used. Figure reproduced from Reference [1] with permission. . . . 21

2.2 PSNR-vs-CR performance of the '24' image from the Kodak dataset for fixed color transforms and the color weight method. (a) Original Image, (b) Rate-Distortion curve. Our method outperforms the baseline transforms. Figure reproduced from Reference [1] with permission. . . . . 26

2.3 PSNR-vs-CR performance of the '23' image from the Kodak dataset for fixed color transforms and the color weight method. (a) Original Image, (b) Rate-Distortion curve. The baseline transforms outperform our method. Figure reproduced from Reference [1] with permission. . . . . 27

2.4 PSNR-vs-CR performance of the 'Lenna' image for fixed color transforms and the color weight method. (a) Original Image, (b) Rate-Distortion curve. Our method outperforms the baseline transforms. Figure reproduced from Reference [1] with permission. 28

2.5 PSNR-vs-CR performance of the '01122182-0033-rgb' image (second image in microscopic image dataset) for fixed color transforms and the color weight method. (a) Original Image, (b) Rate-Distortion curve. Our method outperforms the baseline transforms. Figure reproduced from Reference [1] with permission. . . 29

2.6 A visual result of image '24' from the Kodak dataset coded by JPEG using a quality factor of 80%. (a) Original, (b) JPEG coded version using Y'CbCr and (c) JPEG coded version using the color weight method with Y'CbCr. Figure reproduced from Reference [1] with permission. . . . . 30

2.7 Example of the segmentation using the algorithm described in Chapter 4. (a) shows the original image 1 from the microscopic image dataset and (b) shows the segmentation, where white pixels denote background and black pixels denote foreground. Original image repoduced from Reference [2] with permission. . . . . 37

3.1 Examples of FL images used in this study: (a) FL grade 1, (b) FL grade 2 and (c) FL grade 3 . . . . . 41

3.2 Examples of of the $H_p$ and $E_p$ for the grade 3 image of Figure 3.1 (c): (a) $H_p$, (b) $E_p$ . . . . . 43

3.3 The procedure of the proposed FL classification method. . . . . 44

3.4 Examples of the histograms of $L$ (right) and $H_p$ (left) used in the first step of classification: (a) FL grade 1, (b) FL grade 2, (c) FL grade 3. . . . . 46

3.5 Depiction of the algorithm described in Section 3.2. The histogram in (a) which has empty bins is transformed into (b) using DCT. In (c) the values of the high frequency coefficients are set to zero. The inverse DCT of (c) yields an estimate for the likelihood function that has only non-empty bins, see (d). Note that in (d), bins may have very small likelihood values but larger than zero. Figure reproduced from Reference [3] with permission. . . . . 50

4.1 Estimated $\sigma$ for Sheather's method and the proposed method. . . 67

4.2 Results of KDE for 15 example distributions used in this chapter. Shown are the original (black), KDE with Sheather's method (blue) and KDE with the proposed method cross validation (red) and the proposed cross validation with $l_1$ norm term and filter (green). . . . . 76

5.1 An illustration of the algorithm. The figure shows the final stage of the algorithm. Bin 4 is assigned to cluster U and the algorithm is aborted. . . . . 80

5.2 Example in which our method performed better than Otsu's method according to five out of six metrics: (a) image 6 from the dataset, (b) its groundtruth segmentation, (c) our proposed segmentation and (d) Otsu's segmentation. . . . . 84

5.3 Example in which our method performed worse than Otsu's method according to three out of six metrics: (a) image 2 from the dataset, (b) its groundtruth segmentation, (c) our proposed segmentation and (d) Otsu's segmentation. . . . . 85

5.4 Example in which our method performed worse than Otsu's method according to five out of six metrics: (a) image 47 from the dataset, (b) its groundtruth segmentation, (c) our proposed segmentation and (d) Otsu's segmentation. . . . . 86

5.5 Example of a possible application area: (a) cropped cell image from the BT-20 cancer cell line, (b) our proposed segmentation and (c) Otsu's segmentation. . . . . 87

5.6 Example of a possible application area: (a) cropped cell image from the BT-20 cancer cell line, (b) our proposed segmentation and (c) Otsu's segmentation. . . . . 88

5.7 Example of a possible application area: (a) cropped cell image from a BT-20 cancer cell line, (b) our proposed segmentation and (c) Otsu's segmentation. . . . . 89

5.8 Example of a possible application area: (a) cropped cell image from the BT-20 cancer cell line, (b) our proposed segmentation and (c) Otsu's segmentation. . . . . 90

6.1 Examples of the image classes used in our experiments. Figure reproduced from Reference [4] with permission. . . . . 102

# List of Tables

2.1 The condition numbers of the baseline transforms and the mean and standard deviations of the condition number of our transforms for the Kodak dataset. . . . . 19

2.2 Mean and standard deviations of the correlation coefficients $\rho_{ij}$ for the baseline color transforms and our transforms as computed over the Kodak dataset. . . . . 20

2.3 PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 2.5. . . . . 24

2.4 PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 3. . . . . 25

2.5 PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 2. . . . . 25

2.6 PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. No threshold was used, i.e. the whole image was coded with our method. . . . . 25

2.7 PSNR-Gain values for the whole microscopic image dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 2.5. . . . . 32

2.8 PSNR-Gain values for the whole microscopic image dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 3. . . . . 33

2.9 PSNR-Gain values for the whole microscopic image dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 2. . . . . 34

2.10 PSNR-Gain values for the whole microscopic image dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. No threshold was used, i.e. the whole image was coded with our method. . . . . 35

2.11 PSNR-Gain values for the whole microscopic image dataset using the KDE segmentation of this section. The parameter $a$ was chosen as 0.99. . . . . 38

3.1 Image recognition rates (%) for all three grades and two classifiers different classifiers in the second stage: The proposed sparsity smoothing and SVM. . . . . 51

3.2 Mean and standard deviations of block recognition rates (%) for grades 1 and 2 and two different classifiers in the second stage: The proposed sparsity smoothing and SVM. . . . . 52

3.3 Mean and standard deviations of Kullback-Leibler divergence for test images from Grade 1 to the training images of class A (grade 1 and 2) and class B (grade 3) with $L$ and $H_p$ features. . . . . 52

3.4 Mean and standard deviations of Kullback-Leibler divergence for test images from grade 2 to the training images of class A (grade 1 and 2) and class B (grade 3) with $L$ and $H_p$ features. . . . . 52

3.5 Mean and standard deviations of Kullback-Leibler divergence for test images from Grade 3 to the training images of class A (grade 1 and 2) and class B (grade 3) with $L$ and $H_p$ features. . . . . 53

4.1 KL divergence gain in dB over Sheather's method for traditional cross-validation . . . . . 62

4.2 MISE gain in dB over Sheather's method for traditional cross-validation. . . . . 62

4.3 KL divergence gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. . . . . 63

4.4 MISE gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. . . . . 63

4.5 KL divergence gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_h[n]$ was used. . . . . 64

4.6 MISE gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_h[n]$ was used. . . . . 64

4.7 KL divergence gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_l[n]$ was used. . . . . 65

4.8 MISE gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_l[n]$ was used. . . . . 65

4.9 KL divergence gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_q[n]$ was used. . . . . 66

4.10 MISE gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_q[n]$ was used. . . . . 66

5.1 Differences between Otsu's method and our proposed method for different distance metrics according to Equation (5.7). A positive value indicates that our method had a lower score and was therefore better than Otsu's method. The last row denotes the percentage of images in the dataset where our proposed methods outperforms Otsu's method. . . . . 82

5.2 Differences between Otsu's method and our proposed method for different distance metrics according to Equation (4.14). A positive value indicates that our method had a lower score and was therefore better than Otsu's method. The last row denotes the percentage of images in the dataset where our proposed methods outperforms Otsu's method. . . . . 83

6.1 Covariance vs. co-difference computational cost. The fourth column shows the ratio of the second and third column, respectively. 94

6.2 Classification accuracies (in %) for different classifier inputs. . . . 101

**Dedicated to the memory of Deniz Aksoy.**

# Chapter 1

# INTRODUCTION

> "He is an optician, daily having to deal with the microscope, telescope, and other inventions for sharpening our natural sight, thus enabling us mortals (as I once heard an eccentric put it) liberally to enlarge the field of our essential ignorance."
>
> ***Herman Melville**, Inscription Epistolary to W. Clark Russell*

Vision is arguably one of the most important senses that human beings rely on. However, human vision has only limited resolution and bandwidth. To overcome these limitations has been a steady endeavor of science and engineering from its very beginnings.

The origins of microscopy can be traced back as far as the year 1000 with the first descriptions of concave and convex lenses. In 1595, William Borel delivered the first description of a compound microscope built by the Dutch spectacle-maker Hans Jansen and his son Zacharias. The first detailed scientific study of microscopy dates back to Robert Hooke's milestone book "Micrographia" in 1665 [5]. In 1873, Abbe published his landmark paper [6], finding the relation of the smallest resolvable distance between two points given the wavelength of the imaging light when using a conventional microscope. This diffraction limit

was accepted as the ruling paradigm for imaging until the introduction of sub-diffraction imaging in the early 2000s [7]. Abbe's findings were instrumental in developing the first fluorescent microscope in 1911 by Oskar Heimstaedt [8]. In 1935, Frits Zernike discovered the principle of phase contrast [9]. The advent of the confocal microscope in 1961 [10] allowed researchers to witness a new level of optical resolution. Confocal microscopy can also be used for three-dimensional reconstruction of the sample [11]. Of special interest for image processing practitioners may be the deconvolution-type algorithms in microscopy [12]. An informative overview of the milestones of microscopy is provided in [13].

To illustrate the scope of this thesis, let us first discuss what it is not. This thesis does not deal with the actual image acquisition process of microscopic images. Data was provided by collaborators, e.g., the molecular biology department of Bilkent University, using state-of-the-art imaging systems. The problems that will be adressed in this thesis are therefore issues that arise after imaging and mostly deal with the analysis of the information contained in the images at hand, i.e., the whole processing pipeline of microscopic images, consisting of compression, analysis and classification.

Depending on the sample under observation, we will deal with a large variety of image modalities, from which a variety of image processing problems arise. Example images are shown in Figures 2.2 (a), 3.1 and 6.1. We ask the following question: Can the specific features of microscopic images, such as color content and morphology be used for more efficient compression and classification? The short answer is a resounding "yes". The long answer will be given in the remainder of this thesis.

## 1.2 Microscopic Image Compression

As the amount of digitally stored data is increasing rapidly [14], data compression is a topic that needs to be adressed. The National Institute of Health (NIH) in America issued a call for proposals on the problem of compression of microscopic image data in April 2009. Recent advances in technology have enabled researchers to digitize pathological whole-slides at high magnifications (e.g. at 40x). Resulting color microscopic images are large in size (e.g. 100K-by-100K). Therefore, finding efficient ways of compressing such data is necessary for efficient transmission, storage and evaluation.

Currently, standard image compression methods such as JPEG or JPEG 2000 are used for microscopic image compression. The DCT-based "lossy" JPEG standard [15] has proven to be the most widely accepted technique in modern day applications, thus making the very term JPEG a household name. JPEG, however, was designed and performs best for so-called "natural" images. For example, it uses color transforms of the YUV flavor, in which the luminance component holds most of the energy, compared to the other two chroma components. The weighting coefficients in these color transform matrices model the human visual system (HVS). For other, most notably medical, applications this strict assumption on the color content of the given image may not be valid. This is due to the fact that images of tissue are usually stained with certain dyes [16]. For example, tissue samples showing different follicular lymphoma grades are stained using haematoxylin and eosin (H&E) dyes. Haematoxylin stains cell nuclei blue, while eosin stains cytoplasm, connective tissue and other extracellular substances pink or red. The respective color content of those images is therefore significantly different from the color content of "natural scene" images.

The motivation of our research here is to find an efficient method to represent the color content of a given image in a way that can be integrated with widely

used compression schemes such as JPEG and JPEG 2000. In this thesis, we developed a weight-based color transformation method for microscopic images, that takes the image specific color-content into account and can replace standard color transforms in JPEG and other compression schemes. In the conducted tests the proposed method yielded an increase in PSNR value up to 0.15 dB for natural images and up to 0.5 dB for microscopic images, compared to standard JPEG.

## 1.3 Microscopic Image Analysis

To this day, analysis of microscopic images of human tissues remains the most reliable way of diagnosing and grading several illnesses, e.g., cancer. Our work on the classification and rating of microscopic images is motivated by the fact that there is significant inter- and intra-rater variability in the grading and diagnosis of these illnesses from image data by human experts. Computer-aided classification can assist the experts in making their grading and diagnosis reproducible, while additionally providing useful quantitative measures.

One of the first applications of computer-aided diagnosis (CAD) was digital mammography in the early 1990's [17], [18]. CAD has become a part of routine clinical detection of breast cancer on mammograms at many screening sites and hospitals [19] in the United States. Interest in research of medical imaging and diagnostic radiology has increased since then. Given modern day computing resources, computer-aided image analysis can play its part in facilitating disease classification. Using CAD makes also sense when considering another issue: It has been estimated that roughly 80% of all yearly prostate biopsies in the U.S.A. are benign [20], which means that pathologists spend a larg amount of their time sieving through benign tissue. If CAD could help pathologists to analyse benign tissue faster, a significant amount of time and money could be saved.

The authors of [20] wrote: "Researchers in both the image analysis and pathology fields have recognized the importance of quantitative analysis of pathology images. Since most current pathology diagnosis is based on the subjective (but educated) opinion of pathologists, there is clearly a need for quantitative image-based assessment of digital pathology slides." Applications are in cancer research. Similar techniques can also be used for problems arising in molecular biology.

### 1.3.1 Feature Extraction

The first step of most image interpretation problems including microscopic image processing is feature extraction [21]. Feature vectors represent or summarize a given image. In many problems, image classification is then performed using the feature vectors instead of actual pixel values. Microscopic images are large in size, which requires the feature extraction process to be computationally efficient in order not to rely on supercomputers or computer clusters in the future. Furthermore, the proper choice of feature vectors is crucial.

One problem in this context is not to be underestimated: Humans' concept of the world is inherently object-based, while in image processing we deal with the real world's pixel-based representation. However, human experts describe and understand images in terms of such rather abstract objects. A pathologist, e.g., will describe his or her diagnosis criteria in terms of notions like "nucleus" and "cell." Establishing the link between the "object" and its pixel representation in the image is not straight-forward. Research on useful features for disease classification has often been inspired by visual attributes defined by clinicians as particularly important for disease grading and diagnosis. This ultimately means that the image processing specialist has to learn these features from a seasoned pathologist and thereby become similarily well-skilled in pathology. This may involve a long, iterative process. However, model complexity can not be increased endlessly without the risk of overfitting [22]. One may also ask if it is always

wise to try to mimick the process of a pathologist step-by-step or if a set of low-level features paired with an efficient classifier may result in a better performing sytem. We will introduce such an approach in Chapter 3.

One of the most promising feature extraction methods in recent years has been the region covariance method, which was successfully used in image texture classification and object recognition [23]. Region covariance method features are extensively used in texture classification and recognition in image and video. Given a feature vector of an image region, a covariance matrix is constructed, thereby measuring the interplay of the given features. Covariance matrix computation for $N$ features is of order $\mathcal{O}(N^2)$, due to the extensive number of vector multiplications involved. As mentioned above, the size of microscopic image data requires the feature extraction process to be computationally efficient. The so-called region co-difference matrices serve the same purposes as region covariance matrices and exhibit similar performance but do no rely on the multiplication operator [24]. Texture classification techniques like this can be used for microscopic images to classify them into groups and recognize specific types of diseases. We present frameworks based on these ideas in Chapter 3 (for region covariance) and Chapter 6 (for region co-difference). For a further in-depth review of feature extraction for microscopic images, the reader may consult [20].

## 1.4 Microscopic Image Classification

After feature extraction has been carried out successfully, classification is performed. The problems to be solved can range from binary classification within a given image, e.g., segmentation of pixels into foreground and background, to discrimination of several images into a set of classes. A large amount of literature on classification schemes exists [25]. The Bayes classifier minimises the probability of misclassification [26], but it requires an estimate of the likelihood function

of the data. However, this function is usually unknown. Histograms of the data can be used for this purpose, but due to the limited amount of data, these will be rather poor estimates of the underlying probability density functions (PDFs). In fact, smoothing the histograms will result in more desirable estimates [27]. We will present several algorithms in order to achieve this task.

### 1.4.1 Follicular Lymphoma Classification Using Sparsity Smoothing

The first microscopic image classification problem that we will discuss deals with follicular lymphoma (FL). FL is a group of malignancies of lymphocyte origin that arise from lymph nodes, spleen and bone marrow in the lymphatic system in most cases and is the second most common non-Hodgkin's lymphoma. FL can be differentiated from all other subtypes of lymphoma by the presence of a follicular or nodular pattern of growth presented by follicle center B cells consisting of centrocytes and centroblasts. In practice, FL grading process often depends on the number of centroblasts counted within representative follicles, resulting in three different grades describing the severity of the disease. Although reliable clinical risk stratification tools are available for FL, the optimal choice of treatment continues to depend heavily on morphology-based histological grading.

Current microscopic image classification systems for FL consist of follicle detection using immunohistochemical (IHC) images, image registration, follicle detection using haematoxylin-eosin (H & E) stained images, centroblast detection, and false positive elimination [28]. Therefore, microscopic image analysis for this disease involves many algorithmic components. Optimal and robust functioning of each of these components is important for the overall accuracy of the final system.

In Chapter 3, a different approach in terms of feature extraction and classification was chosen. The philosophy behind this is the following: Features that pathologists and molcecular biologists are interested in with respect to biomedical images are oftentimes hard to model on an image level. If, however, it is known that the feature in question can be decomposed into several low-level components, but their interplay in a particular instance of a feature may be unknown, the covariance of these features may help in classifying the feature correctly.

In Chapter 3, we wish to classify images of the three different grades of FL. In order to carry out classification of the computed features we need to estimate their underlying likelihood functions. Such functions tend to be smooth functions, i.e., they are sparse in the Fourier domain. We introduce a novel classification algorithm that imposes sparsity on the likelihood estimate in an iterative manner. The transform domain in which we will impose sparsity is the discrete cosine transform (DCT), because of its energy compaction property [29]. Once an estimate of the likelihood function is found, a Bayes classifier can be used for the final classification. Classification accuracies for our proposed methods were above 98% for all three FL grades.

### 1.4.2 Bandwidth Selection in Kernel Density Estimation using Fourier Domain Constraints

We believe that using sparsity in PDF estimation can be a powerful tool. In fact a wide range of literature on the issue of sparsity exists, especially within the context of what is commonly referred to as "compressive sensing" (CS) [30]. CS is a technique in which an ill-conditioned system like $Ax = b$, where $A$ does not have full-rank, can be solved uniquely with high probability. This, however, can only be achieved if the data $x$ is sparse in some basis. One can measure sparsity by computing the signal's $l_0$ norm. Unfortunately, minimizing

an expression including the $l_0$ norm is non-deterministic polynomial-time (NP) hard. Therefore, one usually uses the $l_1$ norm as a sparsity restriction for a minimization.

In Chapter 4, we will not use full CS-type algorithms but we will impose sparsity in the Fourier domain on the PDF estimates for kernel density estimation (KDE), via minimization of the $l_1$ norm. In KDE, a PDF can be estimated from data using a kernel function, the most popular of which is the Gaussian kernel. A crucial problem here is how to choose the the bandwidth of the kernel or, in case of the Gaussian kernel, its standard deviation $\sigma$. In case the $\sigma$ is chosen too large or too small, the resultant PDF will be over-smoothed or under-smoothed, respectively. In the statistics community, many algorithms have been developed to estimate a proper bandwidth. The common opinion here is that so-called "plug-into-the-equation" type estimators [31] outperform other approaches [32]. These types of algorithms iteratively update their estimates, but initially rely on an estimate from a parametric family, the so-called "pilot estimate". A concern that is voiced frequently in the statistics community [33] is that the pilot estimate may be computed under wrong assumptions. In this thesis, we are interested in finding approaches to the described problem by using the Fourier domain properties of the data. Similar approaches are to the best of our knowledge absent in the literature. We present an algorithm that outperforms the traditional method by Sheather for different lengths of data. In our simulations the proposed method resulted in Kullback-Leibler divergence gains of up to 2dB.

### 1.4.3 Image Thresholding using Mixtures of Gaussians

Image segmentation is an important part of image understanding. The most basic problem here is segmentation of the pixels into foreground and background. This is usually accomplished by choosing an appropiate threshold [34]. The resultant binary image can be used as input for more advanced segmentation

algorithms [35]. Therefore, the quality of the final segmentation in that case will largely depend on the initial threshold choice, which is why image thresholding can only be considered a "basic" problem at first sight. In microscopic images, thresholding can be used to find the outline of cell organels within the nucleus. Examples are shown in Figures 5.5-5.8.

In order to estimate a threshold in a simple foreground-background image segmentation problem, mixtures of Gaussians can also be used. To this end we present an algorithm that outperforms the widely used method by Otsu [36] in Chapter 5. Image histogram thresholding is carried out using the likelihood of a mixture of Gaussians. In the proposed approach, a PDF of the data is estimated using Gaussian kernel density estimation in an iterative manner, using the method from Chapter 4. The threshold is found by iteratively computing a mixture of Gaussians for the two clusters. This process is aborted when the current bin is assigned to a different cluster than its predecessor. The method is self-adjusting and does not involve an exhaustive search. Visual examples of our segmentation versus Otsu's thresholding method are presented. We evaluated the merit of our method on a dataset of 49 images with groundtruth segmentations. For a given metric our proposed method yielded up to 10% better segmentations than Otsu's method.

### 1.4.4 Cancer Cell Line Classification using Co-Difference Operators

Cancer cell lines (CCL) are widely used for research purposes in laboratories all over the world. They represent different generations of cancer cells that are usually grown in a laboratory environment and whose bio-chemical properties are well known. Molecular biologists use CCLs in their experiments for testing the effects of certain drugs on a given form of cancer. However, during those

experiments the identity of the CCLs have to be confirmed several times. The traditional way of accomplishing this is to use short tandem repeat (STR) analysis. This is a costly process and has to be carried out by an expert. Computer-assisted classification of CCLs can alleviate the burden of manual labeling and help cancer research. In Chapter 6, we present a novel computerized method for cancer cell line image classification.

The aim is to automatically classify 14 different classes of cell lines including 7 classes of breast and 7 classes of liver cancer cells. Microscopic images containing irregular carcinoma cell patterns are represented by subwindows which correspond to foreground pixels. The images of the CCLs contain a large number of different junctions and corners, whose orientation is random. To quantify these directional variations, a region covariance/co-difference descriptor utilizing the dual-tree complex wavelet transform (DT-$\mathbb{C}$WT) coefficients and several morphological attributes was computed. Directionally selective DT-$\mathbb{C}$WT feature parameters are preferred primarily because of their ability to characterize edges at multiple orientations which is the characteristic feature of carcinoma cell line images. A support vector machine (SVM) with radial basis function (RBF) kernel was employed for final classification. The proposed system can be used as a reliable decision maker for laboratory studies. Our tool provides an automated, time- and cost-efficient analysis of cancer cell morphology to classify different cancer cell lines using image-processing techniques, which can be used as an alternative to the costly STR analysis.

In order to achieve this result, a new framework for signal processing is introduced. It is based on a novel vector product definition that permits a multiplier-free implementation. In this framework, a new product of two real numbers is defined as the sum of their absolute values, with the sign determined by the product of the two hard-limited numbers. This new product of real numbers is used to define a similar product of vectors in $R^N$. The new vector product of two

identical vectors reduces to a scaled version of the $l_1$ norm of the vector. The main advantage of this framework is that it yields multiplication-free computationally efficient algorithms for performing some important tasks in microscopic image processing. This is necessary to achieve an almost-real time processing of microscopic images, where computer-aided analysis can provide results within seconds, compared to minutes when a human being carries out the same task. In research environments, a speed-up on such a scale will help medical doctors or molecular biologists significantly. Over a dataset of 280 images, we achieved an accuracy above 98%, which outperforms the classical covariance-based methods.

# Chapter 2

# MICROSCOPIC IMAGE COMPRESSION

Image compression is a well-established and extensively studied field in the signal processing and communication communities. Although the "lossy" JPEG standard [15] is one of the most widely accepted image compression techniques in modern day applications, its resulting fidelity can be improved. It is a well known fact that JPEG compression standard is optimized for natural images. More specifically the color transformation stage is designed in such a way that, it favors the color components to which the human visual system (HSV) is more sensitive in general. However, using one fixed color transformation for all types of natural images or even for all the blocks of an image may not be the most efficient way.

One possible idea is to find a color transform that represents the RGB components in a more efficient manner and can thereby replace the well-known RGB-to-YCbCr or RGB-to-YUV color transforms, used by most practitioners. Usually such approaches aim at reducing the correlation between the color channels [37]. An optimal solution would be to use Karhunen-Loève Transform (KLT), see [38].

However, in KLT there is an underlying wide-sense stationary random process assumption which may not be valid in natural images. This is because autocorrelation values of the image have to be estimated to construct the KLT matrix, since most natural images cannot be considered as wide-sense stationary random processes, due to edges and different objects. A single auto-correlation sequence cannot represent a given image. Another approach to an optimal color space projection on a four-dimensional colorspace was developed in [39]. The following literature survey aims at establishing today's state of the art in the area of microscopic image compression for medical purposes.

Okurma *et al.* [40] studied the characteristics of pathological microscopic images (PMI). Although colorless by nature, sample images are prepared by using special dyes producing color. When comparing these sample images with natural images based on CIE chromaticity diagrams, it was shown that PMI are more concentrated around specific color components than their natural counterparts.

Nakachi *et al.* [41] developed a full-grown coding scheme for PMI. They avoid the effects of chrominance downsampling by using a Karhunen-Loève Transform (KLT), its main advantages being to reduce correlation between RGB color-components and biasing the signal energy between the transformed components. Ladder networks are used to implement KLT. Experiments show that the KL-transformed images have lower entropy than their RGB or YCbCr counterparts. The actual coding process involves a S+P transform followed by a SPIHT coder for each transformed component.

A much larger branch of research activities on efficient coding of medical images involve wavelet approaches to the problem. Ref. [42] offers a clear introduction to the field.

Manduca [43] presented a study comparing the performance of JPEG, standard wavelet (using run-length and arithmetic coding) as well as wavelet transform followed by a SPIHT coding method. For four different images ('Lena', MRI, CT and X-ray) and compression ratios from 10:1 up to 40:1, the RMS errors of the three methods was compared resulting in a distinguishably better performance of SPIHT, followed by standard wavelet compression and lastly normal JPEG.

Arya *et al.* [44] use a vector quantisation approach for coding medical image sequences. Using interframe correlation by applying a 3-D vector scheme yields significantly higher PSNR values than baseline H.261 video codec. Oğuz and Çetin also used video coders to compress MRI images in the past [45]. Since our research is dedicated to single images and not sequences of such, an approach like this was not further investigated.

Gormish *et al.* introduced their CREW compression system in [46]. This system is lossless and idempotent, using a customised reversible color transform, a wavelet transform and a quantization via alignment routine, in which the frequency bands are "aligned" following a selectable rule. CREW has a tagged file format, that also contains information about importance levels of certain regions and not only pixel data. Experimental results show a compression gain of about 10% in Bits/Pixel. Note that this study aims not particularily at medical, but general high quality images.

Zamora *et al.* [47] worked on a particular problem, namely compression of images showing osteoporotic bone. They exploit certain properties of the image data, namely its small variance in color. For that reason run-length coding is a well-suited approach to coding these types of images. Data given by the authors confirms that run-length coding, though quite a simple technique, is competible with adaptive arithmetic coding, which again can achieve entropy values very

close to the input data's entropy. Assuming a six-color model for the input data compression ratios can go up to 100:1 with low computational effort.

A new transform based on the color content of a given image is developed in this chapter. The proposed transform can be used as part of the JPEG image coding standard, as well as part of other image and video coding methods, including the methods described in [48], [49] and [50].

## 2.1 Algorithm

A typical colorspace transform can be represented by a matrix multiplication as follows:

$$\begin{pmatrix} D \\ E \\ F \end{pmatrix} = \mathbf{T} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}, \tag{2.1}$$

where $\mathbf{T} = [t_{ij}]_{3\times 3}$ is the transform matrix, while R, G and B represent the red, green and blue color components of a given pixel, respectively, and D, E, F represent the transformed values, see [51] and [52]. For example, JPEG uses luminance-chrominance type colorspace transforms and chooses the coefficients in $\mathbf{T}$ accordingly. Examples for these include RGB-to-YCbCr [53], as definded in JPEG File Interchange Format (JFIF), as well as RGB-to-YUV and a digital version of RGB-to-Y'CbCr from CCIR 601 Standard that are used in our experiments as baseline color transforms. Their respective transform matrices are given by

$$\mathbf{T'}_{RGB-to-YCbCr} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix}, \tag{2.2}$$

$$\mathbf{T'}_{RGB-to-Y'CbCr} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{pmatrix}, \tag{2.3}$$

and

$$\mathbf{T'}_{RGB-to-YUV} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix}. \tag{2.4}$$

The Y component of the resultant image is usually called the luminance component, carrying most of the information, while the Cb and Cr components, or U V components, respectively, are called the chrominance components.

In our approach, we manipulate the luminance component, while leaving the chrominance components as they are, i.e., only the coefficients in the first row of the **T**-matrix are modified. The second and third rows of the matrix remain unaltered because in natural images, almost all of the image's energy is concentrated in the Y component [54]. As a result, most of the bits are allocated to the Y component. Consider this: The image '01', from the Kodak dataset [55] used in our experiments, is coded with 2.03 bits per pixel (bpp) using standard JPEG with a quality factor of 80%. The PSNR is 33.39 dB. The Y component is coded with 1.76 bpp, while the chrominance components are coded with 0.27 bpp. Similarily, the 'Barbara' image from our expanded dataset is coded with 1.69 bpp and a PSNR of 32.98 dB, when coded with a quality factor of 80%. The Y component is coded with 1.38 bpp, while the chrominance components are coded with 0.31 bpp.

Recent methods of color transform design include [56], [57] and [58], but all of these methods try to optimize their designs over the entire image. However, different parts of a typical natural image may have different color characteristics. To overcome this problem, a block adaptive method taking advantage of the local color features of an image is proposed. In each block of the image, coefficients

of the color transform are determined from the previously compressed neighboring blocks using weighted sums of the RGB pixel values, making the transform specific to that particular block.

We calculate the coefficients $t_{11}$, $t_{12}$, $t_{13}$ of the first row of the color transform matrix, using the color content of the previous blocks in the following manner:

$$t_{11} = \frac{1}{2} \cdot \left( t'_{11} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i,j,1)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{p=1}^{3} \mathbf{I}(k,l,p)} \right), \tag{2.5}$$

$$t_{12} = \frac{1}{2} \cdot \left( t'_{12} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i,j,2)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{p=1}^{3} \mathbf{I}(k,l,p)} \right) \tag{2.6}$$

and

$$t_{13} = \frac{1}{2} \cdot \left( t'_{13} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i,j,3)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{p=1}^{3} \mathbf{I}(k,l,p)} \right), \tag{2.7}$$

where $\mathbf{I}$ denotes a three-dimensional, discrete RGB image composed of the used subimage blocks, which are to be discussed below, $M$ and $N$ denote the number of rows and columns of the subimage block, respectively, and $t'_{1j}$ denotes the element in the 1-st row and the $j$-th column of the 3-by-3 baseline color transform matrix, e.g. RGB-to-YCbCr. Normally, $M$ and $N$ are equal to 8 if only the previous block is used in JPEG coding.

Equations (2.5)-(2.7) have to be computed for each image block, therefore, the proposed transform changes for each block of the image. The extra overhead of encoding the color transform matrix can be easily avoided by borrowing an idea from standard DPCM coding in which predictor coefficients are estimated from encoded signal samples. In other words, there is no need to transmit or store the transform coefficients because they are estimated from previously encoded blocks. However, the specific 3-by-3 color transform matrix for a given block has to be inverted at the decoder. The computational cost for the inversion of a N-by-N matrix is usually given as $\mathcal{O}(N^3)$, however this is valid only in an asymptotic sense. For 3-by-3 matrices, a closed form expression exists, where the inverse can be found using 36 multiplications and 12 additions. In our case where

we only alter the first row of the color transform matrix, this narrows down to 24 multiplications and 6 additions.

Since the color transform matrix is data specific, one may ask how numerically well-conditioned it is. A common technique to measure this is the condition number of a matrix. The condition number is defined as the ratio of the largest to the smallest singular value of the singular value decomposition (SVD) of a given matrix [59]. A condition number with value close to 1 indicates a numerically stable behaviour of the matrix, i.e., it has full rank and is invertible. In order to investigate this, the condition number for each transform matrix of each block of the Kodak dataset was computed. Those results are averaged and can be seen alongside the values of the baseline transform matrices in Table 2.1. We find that for the given dataset, the condition number of our transform is in fact lower than the respective condition number of the baseline transform. It may also be of interest if our modified transform increases the interchannel correlation. In order to investigate this, the correlation coefficients $\rho_{ij}$, denoting the correlation between the $i$-th and $j$-th channel of a color transformed image, were calculated for the baseline transform matrices and for the modified transform matrices over the whole Kodak dataset. The average results can be seen in Table 2.2. We find that for the given dataset, the correlation between channels was not significantly increased by our method.

| Baseline Color Transform | Condition Number Baseline | Condition Number Our Transform |
|---|---|---|
| YCbCr | 1.75 | $1.41 \pm 0.07$ |
| Y'CbCr | 1.75 | $1.38 \pm 0.08$ |
| YUV | 2.00 | $1.72 \pm 0.05$ |

Table 2.1: The condition numbers of the baseline transforms and the mean and standard deviations of the condition number of our transforms for the Kodak dataset.

In most cameras, image blocks are raster-scanned from the sensor and blocks are fed to a JPEG encoder one by one [48]. For the first block of the image, the

| Color Transform | $\rho_{12}$ | $\rho_{13}$ | $\rho_{23}$ |
|---|---|---|---|
| YCbCr | -0.0008 ± 0.2250 | -0.0436 ± 0.1308 | -0.1683 ± 0.2372 |
| Y'CbCr | -0.0006 ± 0.2260 | -0.0444 ± 0.1306 | -0.1691 ± 0.2369 |
| YUV | -0.0008 ± 0.2264 | -0.0444 ± 0.1308 | -0.1696 ± 0.2378 |
| Our YCbCr | 0.0087 ± 0.2255 | 0.0043 ± 0.1308 | -0.1683 ± 0.2337 |
| Our Y'CbCr | 0.0096 ± 0.2263 | 0.0073 ± 0.1306 | -0.1691 ± 0.2337 |
| Our YUV | 0.0087 ± 0.2269 | 0.0041 ± 0.1308 | -0.1696 ± 0.2343 |

Table 2.2: Mean and standard deviations of the correlation coefficients $\rho_{ij}$ for the baseline color transforms and our transforms as computed over the Kodak dataset.

baseline color transform is used and the right-hand side of Equations (2.5)-(2.7) are computed from encoded-decoded color pixel values. For the second image block these color transform coefficients are inserted into the first row of the baseline color transform and it is encoded. The color content of the second block is also computed from encoded-decoded pixel values and used in the coding of the third block. Due to the raster-scanning, the correlation between neighboring blocks is expected to be high, therefore, for a given image block, the color content of its neighboring blocks is assumed to be a good estimate of its own color content. Furthermore, we are not restricted to use a single block to estimate the color transform parameters. We can also use image blocks of previously encoded upper rows as shown in Figure 2.1 in which shaded blocks represent previously encoded blocks and the black shaded block is the current block. The neighboring blocks marked by an arrow are used for the prediction of the current block. In [60] an adaptive scheme is presented in which the encoder selects for each block of the image between the RGB, YCoCg and a simple green, red-difference and blue-difference color spaces. This decision is signaled to the decoder as side information. Our method, however, does not require any transmission of side information to the receiver.

The current block's color content may be significantly different from previously scanned blocks. In such blocks we simply use the baseline color transform.

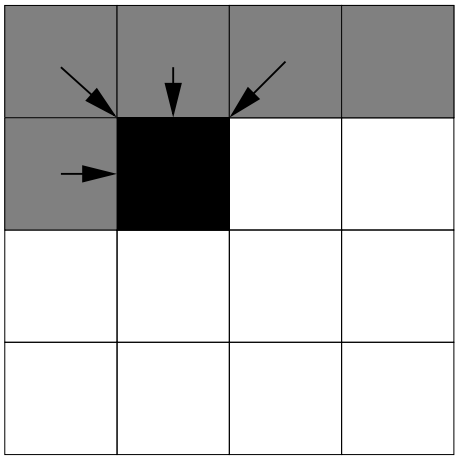


Figure 2.1: A general description of our prediction scheme. To predict the color content of the black-shaded image block, color contents of previously encoded gray-shaded blocks, marked by arrows, are used. Figure reproduced from Reference [1] with permission.

Such a situation may occur if the current block includes an edge. We determine these blocks by comparing the color content with a threshold, as follows

$$\frac{1}{2} \cdot ||\mathbf{x}_c - \overline{\mathbf{x}}_p||_1 > \delta, \tag{2.8}$$

where $\mathbf{x}_c$ is the normalised weight vector of the current block's chrominance channels, $\overline{\mathbf{x}}_p$ is the mean vector of the chrominance channels' weights for all the neighboring blocks used in the prediction and $\delta$ is the similarity threshold. The $l_1$ norm was chosen due to its low computational cost. Note that in our predicition scheme we are not changing the chrominance channels. Therefore, we can use these for estimating the color content of the previous and current blocks, regardless of the changes we make in the luminance channel. The threshold is chosen after investigating the values of the left hand side of Equation 2.8 for the Kodak dataset and calculating its mean and standard deviation. $\delta$ is then chosen according to

$$\delta = \mu_b + \alpha \cdot \sigma_b, \tag{2.9}$$

where $\mu_b$ and $\sigma_b$ denote the mean and standard deviation of the left hand side of Equation 2.8, respectively. $\alpha$ can take values between 2 and 3, since we assume a Gaussian model for the left hand side of Equation 2.8. In a Gaussian distribution,

about 95.4% of the values are within two standard deviations around the mean ($\mu_b \pm 2 \cdot \sigma_b$), and about 99.7% of the values lie within 3 standard deviations around the mean ($\mu_b \pm 3 \cdot \sigma_b$). Therefore, in Equation 2.8 we intend to measure if the $l_1$ norm of the difference between the weight vectors of the current and the previous block lies within an interval of 2-3 standard deviations of the mean value. If it doesn't, we assume that it is an outlier and therefore use the baseline color transform. In Section 2.2 we investigate the performance of several $\alpha$ values.

Due to our prediction scheme, no additional information on the color transform needs to be encoded by implementing a decoder inside the encoder as in standard DPCM signal encoding. It should be also pointed out that optimized color transform designs of [56], [57] and [58] can be also used in our DPCM-like coding strategy. Instead of estimating the color transform over the entire image the transform coefficients can be determined in the previously processed blocks as described above. The goal of this chapter is to introduce the block-adaptive color transform concept within the framework of JPEG and MPEG family of video coding standards. Therefore, a heuristic and computationally simple color transform design approach is proposed in Equations (2.5)-(2.7). Since only the first row of the transform is modified it is possible to use the binary encoding schemes of JPEG and MPEG coders.

## 2.2 Experimental Studies and Results

A dataset of 42 images was used in our experiments. This includes the Kodak dataset, 10 high-resolution images ('1pmw', 'ATI', 'DCTA', 'Gl_Boggs', vHuvahendhoo', 'Patrick', 'PMW', 'LagoonVilla', 'Lake June', 'Sunset Water Suite'), the microscopic image 'Serous-02' [2] and the standard test images 'Lenna', 'Baboon', 'Goldhill', 'Boats', 'Pepper', 'Airplane' and 'Barbara'. The high-resolution images have dimensions ranging from 1650-by-1458 to 2356-by-1579. The JPEG

coder available in the *imwrite* function of MATLAB® was used in our experiments. The color transformation stage of the baseline JPEG was replaced with the proposed form of transformation. The weights of Equations (2.5)-(2.7) were computed using the previously processed blocks neighboring the current block as shown in Figure 2.1.

We show several tables, in which we alter the $\alpha$ value of Equation 2.9. We chose $\alpha$ to be 2, 2.5 and 3, as explained in Section 2.1. The results can be seen in Tables 2.3-2.5. The results for using no threshold at all, i.e., the whole image being coded by our method, can be seen in Table 2.6. Note that the $\delta$ threshold from Equation 2.8 was computed using the data from the Kodak dataset but still performs well on the 18 additional images.

The PSNR-Gain of our method over the baseline color transform was measured at five different compression ratios (CR), spread over the whole rate range, for each image. The averages of these gain values are shown in the tables. Additionally, the mean of all these gain values is presented for the whole dataset. Furthermore, a success rate for the dataset is given. The decision for a success is binary and is made in case the average gain value of a given image is positive. These results show that, on average, the proposed method produces better results than the baseline JPEG algorithm using the RGB-to-YCbCr, RGB-to-YUV or RGB-to-Y'CbCr matrices, respectively. Using the threshold from Equation 2.8 yields better results than using no theshold. On average, our method used with Y'CbCr yields the best compression gain. This tendency can be seen in most images. The two images were the performance of our method is the worst are images '3' and '23'. In those cases, Y'CbCr has a positive gain but significantly smaller than its average gain, thus a negative tendency on the coding performance in those images is perceivable for all color spaces. Especially in those images, many sharp edges are visible and the color content on one side of the edge is not highly correlated to the color content on the other side of the edge. The differences in

| Image | Average PSNR Gain [dB] using YCbCr as baseline | Average PSNR Gain [dB] using Y'CbCr as baseline | Average PSNR Gain [dB] using YUV as baseline |
|---|---|---|---|
| '1' | 0.0624 | 0.0928 | 0.0732 |
| '2' | -0.0668 | -0.0845 | -0.0368 |
| '3' | -0.2394 | 0.0824 | -0.6358 |
| '4' | -0.0325 | 0.0017 | -0.2449 |
| '5' | 0.0423 | 0.1008 | 0.0863 |
| '6' | 0.0966 | 0.1302 | 0.1564 |
| '7' | 0.0480 | 0.0767 | 0.0115 |
| '8' | 0.0958 | 0.1237 | 0.1326 |
| '9' | 0.1147 | 0.1349 | 0.1868 |
| '10' | 0.1395 | 0.2309 | 0.2167 |
| '11' | 0.0300 | 0.0791 | 0.0263 |
| '12' | 0.0781 | 0.0534 | 0.1261 |
| '13' | 0.1179 | 0.1162 | 0.1236 |
| '14' | -0.0517 | -0.0901 | -0.0538 |
| '15' | -0.0518 | -0.0486 | 0.0024 |
| '16' | 0.0812 | 0.1545 | 0.1415 |
| '17' | 0.0845 | 0.1265 | 0.1553 |
| '18' | 0.0952 | 0.1283 | 0.1113 |
| '19' | 0.0500 | 0.1003 | 0.0947 |
| '20' | 0.0399 | 0.0581 | 0.1320 |
| '21' | 0.0799 | 0.1535 | 0.1305 |
| '22' | 0.0642 | 0.1371 | 0.0762 |
| '23' | -0.4293 | 0.0525 | -1.1956 |
| '24' | 0.1448 | 0.1903 | 0.2081 |
| '1pmw' | 0.1832 | 0.1659 | 0.2331 |
| 'ATI' | 0.0289 | 0.1388 | 0.1586 |
| 'Airplane' | 0.5197 | 0.5079 | 0.4287 |
| 'Baboon' | 0.0003 | 0.2097 | -0.4955 |
| 'Barbara' | 0.1054 | 0.1294 | 0.1155 |
| 'Boats' | 0.0913 | 0.0840 | 0.1348 |
| 'DCTA' | 0.2134 | 0.2063 | 0.2457 |
| 'Gl_Boggs' | 0.4427 | 0.3519 | 0.4673 |
| 'Goldhill' | 0.2324 | 0.2395 | 0.2485 |
| 'Huvahendhoo' | 0.2076 | 0.2254 | 0.2698 |
| 'LagoonVilla' | 0.0791 | 0.0551 | 0.1229 |
| 'Lake June' | 0.1223 | 0.1211 | 0.1388 |
| 'Lenna' | 0.2070 | 0.2472 | 0.2197 |
| 'Patrick' | 0.1130 | 0.0778 | 0.1489 |
| 'Pepper' | 0.2158 | 0.2130 | 0.1769 |
| 'PMW' | 0.1696 | 0.2188 | 0.2078 |
| 'Serous-02' | 0.1245 | 0.0917 | 0.1606 |
| 'Sunset Water Suite' | 0.2036 | 0.3482 | 0.7866 |
| Whole dataset | 0.0967 | 0.1435 | 0.0952 |
| Success rate | 36/42 | 39/42 | 35/42 |

Table 2.3: PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 2.5.

| Image | Average PSNR Gain [dB] using YCbCr as baseline | Average PSNR Gain [dB] using Y'CbCr as baseline | Average PSNR Gain [dB] using YUV as baseline |
|---|---|---|---|
| Whole dataset | 0.0948 | 0.1432 | 0.0941 |
| Success Rate | 36/42 | 39/42 | 35/42 |

Table 2.4: PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 3.

| Image | Average PSNR Gain [dB] using YCbCr as baseline | Average PSNR Gain [dB] using Y'CbCr as baseline | Average PSNR Gain [dB] using YUV as baseline |
|---|---|---|---|
| Whole dataset | 0.0967 | 0.1411 | 0.0965 |
| Success Rate | 35/42 | 39/42 | 35/42 |

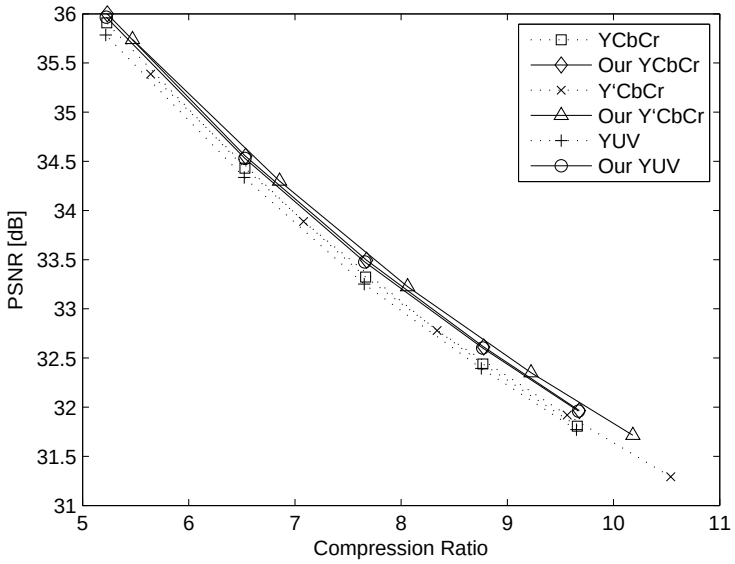
Table 2.5: PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 2.

| Image | Average PSNR Gain [dB] using YCbCr as baseline | Average PSNR Gain [dB] using Y'CbCr as baseline | Average PSNR Gain [dB] using YUV as baseline |
|---|---|---|---|
| Whole dataset | 0.0609 | 0.1207 | 0.0521 |
| Success Rate | 31/42 | 34/42 | 31/42 |

Table 2.6: PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. No threshold was used, i.e. the whole image was coded with our method.

color content of certain blocks compared to their previous blocks are not as well detected by the threshold approach of Section 2.1 as in most other images of our dataset. The color estimate is therefore not accurate, resulting in a larger coding error.

(a)

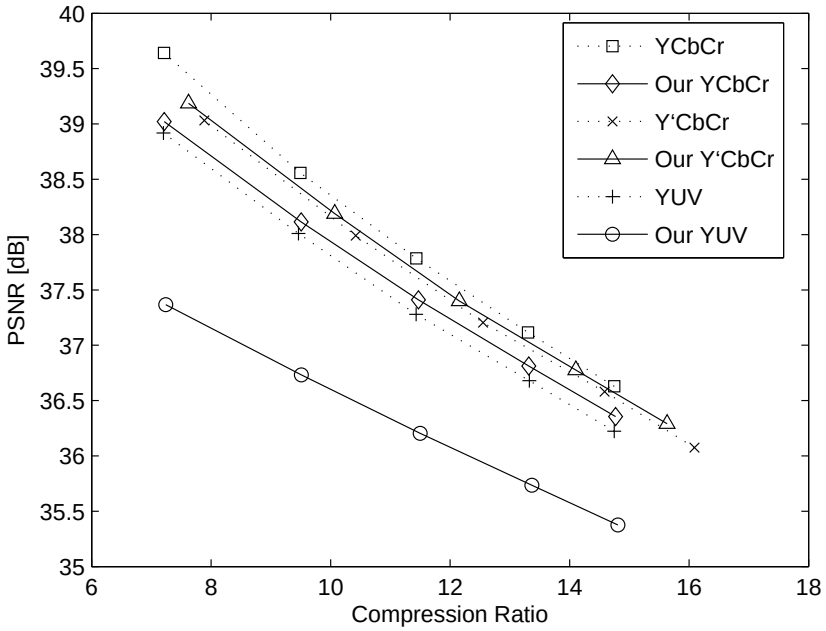



(b)

Figure 2.2: PSNR-vs-CR performance of the '24' image from the Kodak dataset for fixed color transforms and the color weight method. (a) Original Image, (b) Rate-Distortion curve. Our method outperforms the baseline transforms. Figure reproduced from Reference [1] with permission.

In Figures 2.2 - 2.2 the rate-distortion curves for '24', '23', 'Serous-02' and 'Lenna' are given. While v24', 'Serous 2' and 'Lenna' are images where our method outperforms the baseline transforms, in image '23' this is not the case.

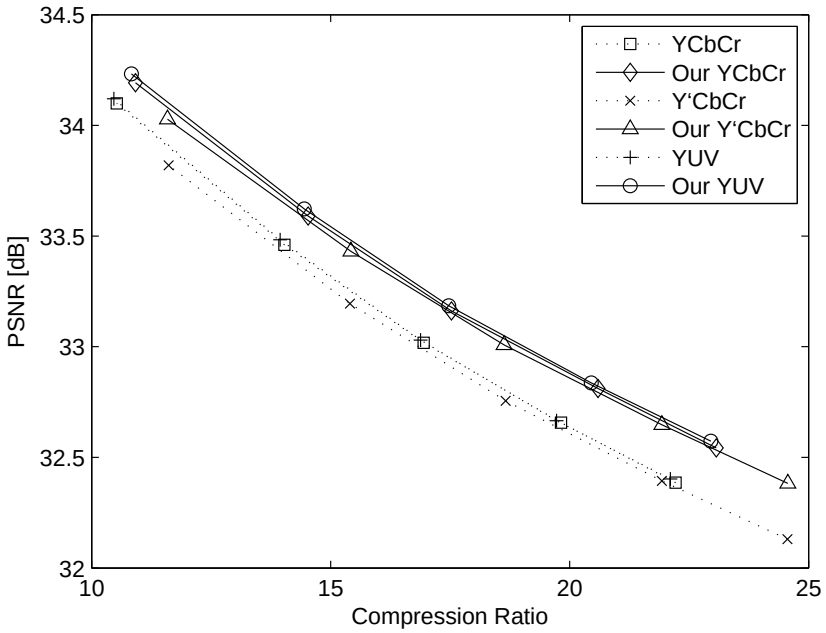
(a)




(b)

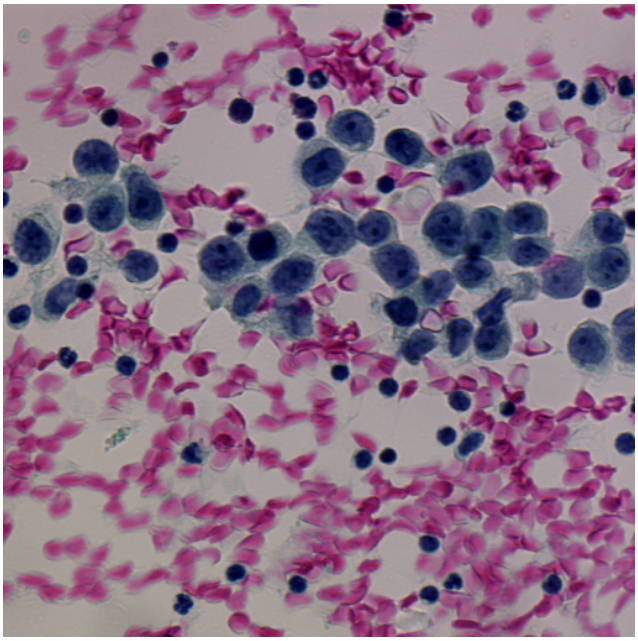
Figure 2.3: PSNR-vs-CR performance of the '23' image from the Kodak dataset for fixed color transforms and the color weight method. (a) Original Image, (b) Rate-Distortion curve. The baseline transforms outperform our method. Figure reproduced from Reference [1] with permission.

(a)

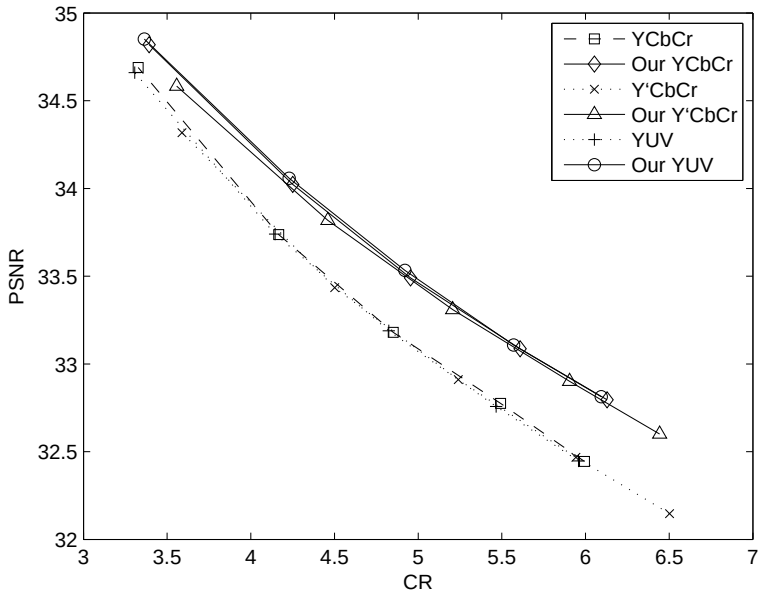



(b)

Figure 2.4: PSNR-vs-CR performance of the 'Lenna' image for fixed color transforms and the color weight method. (a) Original Image, (b) Rate-Distortion curve. Our method outperforms the baseline transforms. Figure reproduced from Reference [1] with permission.

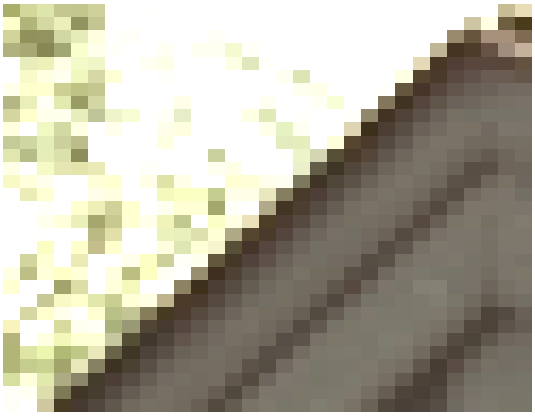


(a)




(b)

Figure 2.5: PSNR-vs-CR performance of the '01122182-0033-rgb' image (second image in microscopic image dataset) for fixed color transforms and the color weight method. (a) Original Image, (b) Rate-Distortion curve. Our method outperforms the baseline transforms. Figure reproduced from Reference [1] with permission.



(a)

(b)

(c)

Figure 2.6: A visual result of image '24' from the Kodak dataset coded by JPEG using a quality factor of 80%. (a) Original, (b) JPEG coded version using Y'CbCr and (c) JPEG coded version using the color weight method with Y'CbCr. Figure reproduced from Reference [1] with permission.

Images with strong, saturated color content that changes abruptly seem to perform worse with our method than with the baseline transform.

In Figure 2.2, a visual example of our coding results is given. Figure 2.2 (a) shows the original cropped image, while Figures 2.2 (b)-(c) show the coded versions using Y'CbCR and our method based on Y'CbCr, respectively.

Similar experiments were also carried out for a dataset consisting of 51 microscopic images. This dataset was taken from [2]. Cells are colored by the international standard of coloration of [61]. Therefore, it has a limited color content, which is different from the color content of natural images as used in the previous experiment. The dataset consists of eight bronchial images, ten serous images and 33 different cell images. The results can be seen in Tables 2.7 - 2.10. Our method provides a significant gain over the standard color transform, up to an average 0.5 dB.

It is possible to devise a compression scheme for microscopic images by using image segmentation. In JPEG compression one can use a quality factor, which ranges from 0 to 100, where 100 denotes lossless compression and all other values below denote lossy compression where image fidelity is decreasing with the value of the quality factor. In microscopic images of Serous data [2], a large number of the images show a lot of background. Allocating a lot of bits to these image regions is a waste of ressources, since these regions are of no interest for the practitioner. One may decrease the quality factor for these regions while increasing the quality factor for the foreground regions. For example one may choose

$$q_b = a \cdot q_n \tag{2.10}$$

and

$$q_f = (2 - a) \cdot q_n, \tag{2.11}$$

where $q_b$ and $q_f$ denote the quality factors of the background and foreground, respectively and $q_n$ denotes the quality factor that would be chosen if one would

| Image | Average PSNR Gain [dB] using YCbCr as baseline | Average PSNR Gain [dB] using Y'CbCr as baseline | Average PSNR Gain [dB] using YUV as baseline |
|---|---|---|---|
| 1 | 0.1816 | 0.2523 | 0.2952 |
| 2 | 0.1204 | 0.1864 | 0.1973 |
| 3 | 0.2131 | 0.3218 | 0.3051 |
| 4 | 0.1244 | 0.1980 | 0.2044 |
| 5 | 0.1165 | 0.1604 | 0.2076 |
| 6 | 0.1265 | 0.1709 | 0.2016 |
| 7 | 0.1787 | 0.2146 | 0.2701 |
| 8 | 0.1236 | 0.1319 | 0.2228 |
| 9 | 0.4874 | 0.4298 | 0.5057 |
| 10 | 0.4907 | 0.5000 | 0.5239 |
| 11 | 0.3927 | 0.4057 | 0.5843 |
| 12 | 0.3974 | 0.3729 | 0.4475 |
| 13 | 0.4249 | 0.4462 | 0.6996 |
| 14 | 0.3707 | 0.3803 | 0.6418 |
| 15 | 0.6498 | 0.6400 | 0.7446 |
| 16 | 0.3937 | 0.3741 | 0.4524 |
| 17 | 0.3959 | 0.3873 | 0.4759 |
| 18 | 0.2574 | 0.2058 | 0.7414 |
| 19 | 0.2710 | 0.2740 | 0.7392 |
| 20 | 0.2261 | 0.2777 | 0.3289 |
| 21 | 0.2992 | 0.3228 | 0.6336 |
| 22 | 0.3054 | 0.3027 | 0.7169 |
| 23 | 0.1910 | 0.1753 | 0.4611 |
| 24 | 0.2684 | 0.1934 | 0.4788 |
| 25 | 0.2620 | 0.2586 | 0.5628 |
| 26 | 0.4066 | 0.3756 | 0.6622 |
| 27 | 0.3775 | 0.3933 | 0.6911 |
| 28 | 0.3665 | 0.3208 | 0.6472 |
| 29 | 0.3276 | 0.3180 | 0.5650 |
| 30 | 0.4576 | 0.4737 | 0.6702 |
| 31 | 0.3643 | 0.3205 | 0.5450 |
| 32 | 0.2863 | 0.2662 | 0.7242 |
| 33 | 0.5397 | 0.5667 | 0.5864 |
| 34 | 0.3039 | 0.3349 | 0.3651 |
| 35 | 0.4152 | 0.4698 | 0.4952 |
| 36 | 0.5036 | 0.5409 | 0.5419 |
| 37 | 0.5002 | 0.5070 | 0.5372 |
| 38 | 0.6791 | 0.6483 | 0.7942 |
| 39 | 0.4365 | 0.3987 | 0.5008 |
| 40 | 0.4904 | 0.5093 | 0.6956 |
| 41 | 0.3485 | 0.3439 | 0.7660 |
| 42 | 0.4446 | 0.4650 | 0.4905 |
| 43 | 0.3304 | 0.3866 | 0.3665 |
| 44 | 0.3475 | 0.4062 | 0.3718 |
| 45 | 0.4216 | 0.4254 | 0.4545 |
| 46 | 0.4259 | 0.4373 | 0.4927 |
| 47 | 0.3065 | 0.2638 | 0.3636 |
| 48 | 0.3452 | 0.3383 | 0.3959 |
| 49 | 0.1396 | 0.0571 | 0.2289 |
| 50 | 0.1949 | 0.1391 | 0.2679 |
| 51 | 0.3026 | 0.3551 | 0.3769 |
| Whole dataset | 0.3398 | 0.3460 | 0.4949 |
| Success rate | 51/51 | 51/51 | 51/51 |

Table 2.7: PSNR-Gain values for the whole microscopic image dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 2.5.

| Image | Average PSNR Gain [dB] using YCbCr as baseline | Average PSNR Gain [dB] using Y'CbCr as baseline | Average PSNR Gain [dB] using YUV as baseline |
|---|---|---|---|
| 1 | 0.1889 | 0.2586 | 0.3065 |
| 2 | 0.1236 | 0.1893 | 0.2048 |
| 3 | 0.2246 | 0.3333 | 0.3192 |
| 4 | 0.1289 | 0.2077 | 0.2099 |
| 5 | 0.1125 | 0.1603 | 0.2061 |
| 6 | 0.1262 | 0.1718 | 0.2016 |
| 7 | 0.1787 | 0.2185 | 0.2734 |
| 8 | 0.1268 | 0.1326 | 0.2294 |
| 9 | 0.4874 | 0.4298 | 0.5098 |
| 10 | 0.4916 | 0.5000 | 0.5270 |
| 11 | 0.3927 | 0.4057 | 0.5843 |
| 12 | 0.3995 | 0.3729 | 0.4493 |
| 13 | 0.4435 | 0.4558 | 0.7155 |
| 14 | 0.3818 | 0.3793 | 0.6566 |
| 15 | 0.6500 | 0.6401 | 0.7514 |
| 16 | 0.3855 | 0.3792 | 0.4575 |
| 17 | 0.3977 | 0.3913 | 0.4945 |
| 18 | 0.2757 | 0.2105 | 0.7676 |
| 19 | 0.2760 | 0.2687 | 0.7546 |
| 20 | 0.2593 | 0.3132 | 0.3434 |
| 21 | 0.3045 | 0.3365 | 0.6433 |
| 22 | 0.3034 | 0.3045 | 0.7268 |
| 23 | 0.2039 | 0.1597 | 0.4708 |
| 24 | 0.3013 | 0.2221 | 0.5519 |
| 25 | 0.2816 | 0.2648 | 0.5940 |
| 26 | 0.4040 | 0.3749 | 0.6743 |
| 27 | 0.3670 | 0.4002 | 0.6677 |
| 28 | 0.3630 | 0.3277 | 0.6699 |
| 29 | 0.3231 | 0.3463 | 0.5740 |
| 30 | 0.4745 | 0.4857 | 0.7254 |
| 31 | 0.4083 | 0.3440 | 0.6356 |
| 32 | 0.3267 | 0.2837 | 0.7708 |
| 33 | 0.5437 | 0.5667 | 0.5937 |
| 34 | 0.3039 | 0.3349 | 0.3675 |
| 35 | 0.4152 | 0.4698 | 0.4952 |
| 36 | 0.5036 | 0.5409 | 0.5419 |
| 37 | 0.5039 | 0.5101 | 0.5387 |
| 38 | 0.6857 | 0.6483 | 0.8098 |
| 39 | 0.4357 | 0.4093 | 0.5039 |
| 40 | 0.4901 | 0.5099 | 0.7020 |
| 41 | 0.3477 | 0.3486 | 0.7725 |
| 42 | 0.4547 | 0.4784 | 0.5029 |
| 43 | 0.3412 | 0.3991 | 0.3867 |
| 44 | 0.3603 | 0.4194 | 0.3892 |
| 45 | 0.4430 | 0.4452 | 0.4823 |
| 46 | 0.4319 | 0.4422 | 0.5004 |
| 47 | 0.3146 | 0.2701 | 0.3706 |
| 48 | 0.3676 | 0.3610 | 0.4186 |
| 49 | 0.1414 | 0.0572 | 0.2304 |
| 50 | 0.1988 | 0.1391 | 0.2718 |
| 51 | 0.3196 | 0.3658 | 0.3955 |
| Whole dataset | 0.3474 | 0.3526 | 0.5086 |
| Success rate | 51/51 | 51/51 | 51/51 |

Table 2.8: PSNR-Gain values for the whole microscopic image dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 3.

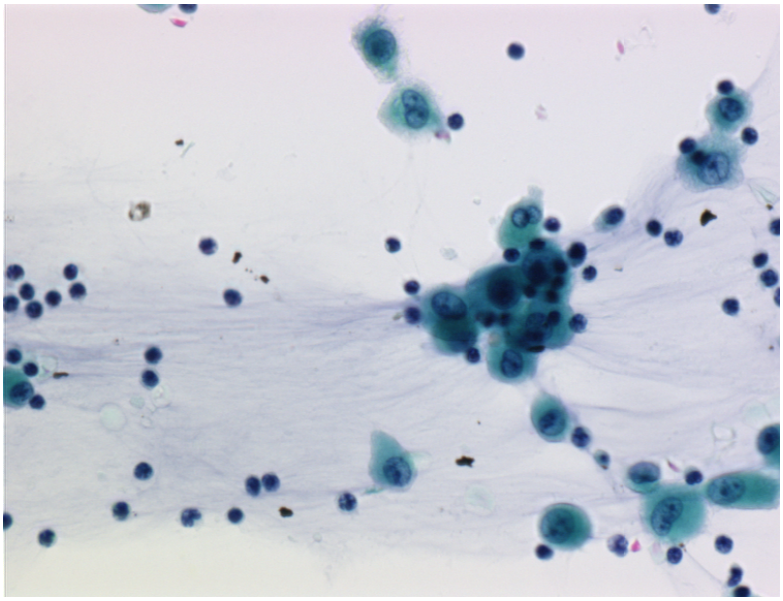
| Image | Average PSNR Gain [dB] using YCbCr as baseline | Average PSNR Gain [dB] using Y'CbCr as baseline | Average PSNR Gain [dB] using YUV as baseline |
|---|---|---|---|
| 1 | 0.1747 | 0.2398 | 0.2783 |
| 2 | 0.1201 | 0.1846 | 0.1910 |
| 3 | 0.2051 | 0.3084 | 0.2919 |
| 4 | 0.1228 | 0.1911 | 0.1908 |
| 5 | 0.1200 | 0.1632 | 0.2077 |
| 6 | 0.1250 | 0.1717 | 0.1987 |
| 7 | 0.1765 | 0.2121 | 0.2683 |
| 8 | 0.1197 | 0.1278 | 0.2173 |
| 9 | 0.4780 | 0.4298 | 0.4990 |
| 10 | 0.4912 | 0.4967 | 0.5020 |
| 11 | 0.3927 | 0.4057 | 0.5760 |
| 12 | 0.3886 | 0.3697 | 0.4456 |
| 13 | 0.4133 | 0.4172 | 0.6897 |
| 14 | 0.3623 | 0.3534 | 0.6343 |
| 15 | 0.6442 | 0.6371 | 0.7332 |
| 16 | 0.3881 | 0.3709 | 0.4317 |
| 17 | 0.3674 | 0.3705 | 0.4652 |
| 18 | 0.2391 | 0.1919 | 0.7304 |
| 19 | 0.2563 | 0.2674 | 0.7036 |
| 20 | 0.2030 | 0.2580 | 0.3174 |
| 21 | 0.2822 | 0.3154 | 0.5947 |
| 22 | 0.2912 | 0.2943 | 0.6845 |
| 23 | 0.1981 | 0.1756 | 0.4296 |
| 24 | 0.2433 | 0.1651 | 0.4466 |
| 25 | 0.2437 | 0.2230 | 0.5292 |
| 26 | 0.4024 | 0.3708 | 0.6454 |
| 27 | 0.3819 | 0.4050 | 0.6839 |
| 28 | 0.3530 | 0.3220 | 0.6094 |
| 29 | 0.3234 | 0.3202 | 0.5152 |
| 30 | 0.3754 | 0.4365 | 0.5915 |
| 31 | 0.3102 | 0.2363 | 0.4636 |
| 32 | 0.2731 | 0.2395 | 0.7007 |
| 33 | 0.5290 | 0.5548 | 0.5821 |
| 34 | 0.2931 | 0.3264 | 0.3517 |
| 35 | 0.4152 | 0.4698 | 0.4952 |
| 36 | 0.5036 | 0.5409 | 0.5319 |
| 37 | 0.5027 | 0.5043 | 0.5395 |
| 38 | 0.6392 | 0.6104 | 0.7430 |
| 39 | 0.4056 | 0.3862 | 0.4618 |
| 40 | 0.4839 | 0.5051 | 0.6879 |
| 41 | 0.3587 | 0.3379 | 0.7645 |
| 42 | 0.4251 | 0.4481 | 0.4671 |
| 43 | 0.3169 | 0.3720 | 0.3429 |
| 44 | 0.3361 | 0.3911 | 0.3511 |
| 45 | 0.3836 | 0.3946 | 0.4157 |
| 46 | 0.4152 | 0.4286 | 0.4753 |
| 47 | 0.3006 | 0.2538 | 0.3513 |
| 48 | 0.3207 | 0.3092 | 0.3677 |
| 49 | 0.1368 | 0.0549 | 0.2265 |
| 50 | 0.1941 | 0.1359 | 0.2652 |
| 51 | 0.2900 | 0.3288 | 0.3543 |
| Whole dataset | 0.3278 | 0.3338 | 0.4753 |
| Success rate | 51/51 | 51/51 | 51/51 |

Table 2.9: PSNR-Gain values for the whole microscopic image dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. $\alpha$ is equal to 2.

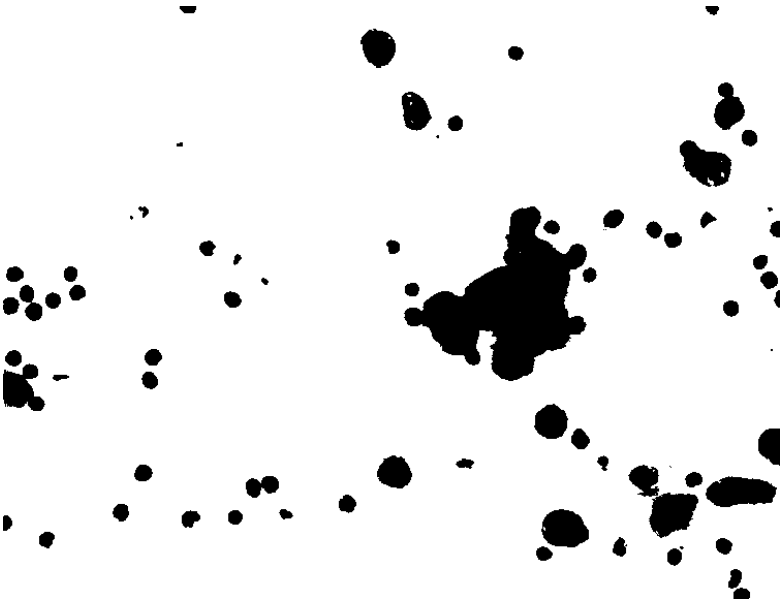
| Image | Average PSNR Gain [dB] using YCbCr as baseline | Average PSNR Gain [dB] using Y'CbCr as baseline | Average PSNR Gain [dB] using YUV as baseline |
|---|---|---|---|
| 1 | 0.1948 | 0.2631 | 0.3185 |
| 2 | 0.1295 | 0.1990 | 0.2210 |
| 3 | 0.2398 | 0.3411 | 0.3418 |
| 4 | 0.1372 | 0.2152 | 0.2288 |
| 5 | 0.1084 | 0.1632 | 0.1966 |
| 6 | 0.1254 | 0.1728 | 0.2050 |
| 7 | 0.1777 | 0.2183 | 0.2799 |
| 8 | 0.1247 | 0.1326 | 0.2293 |
| 9 | 0.4874 | 0.4298 | 0.5098 |
| 10 | 0.4916 | 0.5000 | 0.5270 |
| 11 | 0.3927 | 0.4057 | 0.5843 |
| 12 | 0.3995 | 0.3729 | 0.4509 |
| 13 | 0.4571 | 0.4588 | 0.7432 |
| 14 | 0.3835 | 0.3793 | 0.6724 |
| 15 | 0.6486 | 0.6401 | 0.7501 |
| 16 | 0.3950 | 0.3792 | 0.4757 |
| 17 | 0.4158 | 0.4005 | 0.5210 |
| 18 | 0.2843 | 0.2146 | 0.7981 |
| 19 | 0.2697 | 0.2647 | 0.7478 |
| 20 | 0.3060 | 0.3443 | 0.4281 |
| 21 | 0.2892 | 0.3387 | 0.6586 |
| 22 | 0.3175 | 0.3164 | 0.7390 |
| 23 | 0.1979 | 0.1647 | 0.4716 |
| 24 | 0.3333 | 0.2256 | 0.5854 |
| 25 | 0.2766 | 0.2764 | 0.6189 |
| 26 | 0.4026 | 0.3759 | 0.6786 |
| 27 | 0.3783 | 0.3972 | 0.6790 |
| 28 | 0.3390 | 0.3112 | 0.6582 |
| 29 | 0.3513 | 0.3510 | 0.6334 |
| 30 | 0.5010 | 0.4857 | 0.7579 |
| 31 | 0.4328 | 0.3450 | 0.6722 |
| 32 | 0.3445 | 0.2997 | 0.8111 |
| 33 | 0.5437 | 0.5667 | 0.5983 |
| 34 | 0.3039 | 0.3349 | 0.3675 |
| 35 | 0.4152 | 0.4698 | 0.4952 |
| 36 | 0.5037 | 0.5423 | 0.5437 |
| 37 | 0.5039 | 0.5101 | 0.5422 |
| 38 | 0.6857 | 0.6483 | 0.8118 |
| 39 | 0.4569 | 0.4111 | 0.5250 |
| 40 | 0.4894 | 0.5099 | 0.7002 |
| 41 | 0.3500 | 0.3486 | 0.7814 |
| 42 | 0.4815 | 0.4968 | 0.5344 |
| 43 | 0.3656 | 0.4259 | 0.4247 |
| 44 | 0.3875 | 0.4443 | 0.4307 |
| 45 | 0.4798 | 0.4751 | 0.5299 |
| 46 | 0.4372 | 0.4437 | 0.5090 |
| 47 | 0.3343 | 0.2883 | 0.3948 |
| 48 | 0.4074 | 0.3918 | 0.4717 |
| 49 | 0.1414 | 0.0572 | 0.2316 |
| 50 | 0.1961 | 0.1365 | 0.2711 |
| 51 | 0.3310 | 0.3718 | 0.4093 |
| Whole dataset | 0.3558 | 0.3580 | 0.5248 |
| Success rate | 51/51 | 51/51 | 51/51 |

Table 2.10: PSNR-Gain values for the whole microscopic image dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. No threshold was used, i.e. the whole image was coded with our method.

compresss all image regions with the same quality factor. $a$ is a parameter that can have values ranging from 0 to 1. Figure 2.2 shows the example image 1 from our microscopic image data set and its segmentation based on the threshold algorithm that will be developed in Chapter 4. Table 2.11 shows the averaged results for our dataset. The results are inferior to the adaptive color transform scheme presented above, however it is better on average than standard JPEG and relies on only a very simple pre-processing step, i.e., to find the threshold. If the average of a given block is below that threshold it is considered as foreground and compressed accordingly. Similarly, it is considered as background if the mean value of the given block is above the threshold. The thresholding must perform very well for this method to work. The images in our dataset show objects in front of a white background where the objects are usually significantly darker than the background. Therefore, these images are relatively easy to segment. For other types of images more elaborate pre-processing steps may be necessary.



(a)



(b)

Figure 2.7: Example of the segmentation using the algorithm described in Chapter 4. (a) shows the original image 1 from the microscopic image dataset and (b) shows the segmentation, where white pixels denote background and black pixels denote foreground. Original image repoduced from Reference [2] with permission.

| Image | Average PSNR Gain [dB] using KDE segmentation |
|---|---|
| 1 | 0.0338 |
| 2 | -0.0321 |
| 3 | 0.0487 |
| 4 | -0.0215 |
| 5 | -0.0092 |
| 6 | -0.0098 |
| 7 | 0.0234 |
| 8 | -0.0039 |
| 9 | 0.0972 |
| 10 | 0.1149 |
| 11 | 0.0234 |
| 12 | 0.1007 |
| 13 | 0.1364 |
| 14 | 0.1092 |
| 15 | 0.1604 |
| 16 | 0.0405 |
| 17 | 0.0757 |
| 18 | 0.0567 |
| 19 | 0.0318 |
| 20 | 0.0410 |
| 21 | 0.0432 |
| 22 | 0.0766 |
| 23 | -0.0832 |
| 24 | 0.0916 |
| 25 | 0.0258 |
| 26 | 0.1282 |
| 27 | 0.0047 |
| 28 | 0.0480 |
| 29 | 0.0057 |
| 30 | 0.1594 |
| 31 | 0.1417 |
| 32 | 0.0590 |
| 33 | 0.0840 |
| 34 | -0.0056 |
| 35 | -0.0101 |
| 36 | 0.0812 |
| 37 | 0.0831 |
| 38 | 0.1628 |
| 39 | 0.0291 |
| 40 | 0.2056 |
| 41 | 0.1051 |
| 42 | 0.1383 |
| 43 | -0.0715 |
| 44 | -0.0248 |
| 45 | 0.2101 |
| 46 | 0.1403 |
| 47 | 0.1151 |
| 48 | 0.1324 |
| 49 | 0.0033 |
| 50 | 0.0109 |
| 51 | 0.1088 |
| Mean | 0.0631 |
| Success rate | 41/51 |

Table 2.11: PSNR-Gain values for the whole microscopic image dataset using the KDE segmentation of this section. The parameter $a$ was chosen as 0.99.

# Chapter 3

# CLASSIFICATION OF FOLLICULAR LYMPHOMA IMAGES BY REGION COVARIANCE AND A BAYESIAN CLASSIFIER USING SPARSITY IN TRANSFORM DOMAIN

Follicular lymphoma (FL) is the second most common form of non-Hodgkin lymphoma. It affects mainly adult patients in the western world and represents 20-25% of non-Hodgkin lymphomas in the United States [62]. According to [63]: "FL is a follicular center B cell derived malignancy that is characterized by typical morphologic, immunophenotypic and cytogenetic findings. Classical

follicular pattern results from accumulation of malignant B-lymphocytes including small-cleaved cells (centrocytes), and large non-cleaved cells (centroblasts). Histological grading of FL is critical for proper risk stratification and is used to guide choice of optimal therapy for FL patients. Histological grading is based on the average number of centroblasts manually counted by the pathologist in ten random standard microscopic high power fields (40x) representing neoplastic follicles. This manual method even when performed by expert hematopathologist has well documented high inter- and intra-observer variability leading to poor reproducibility of FL grading."

The world health organization (WHO) defines three different gradings for FL [64]:

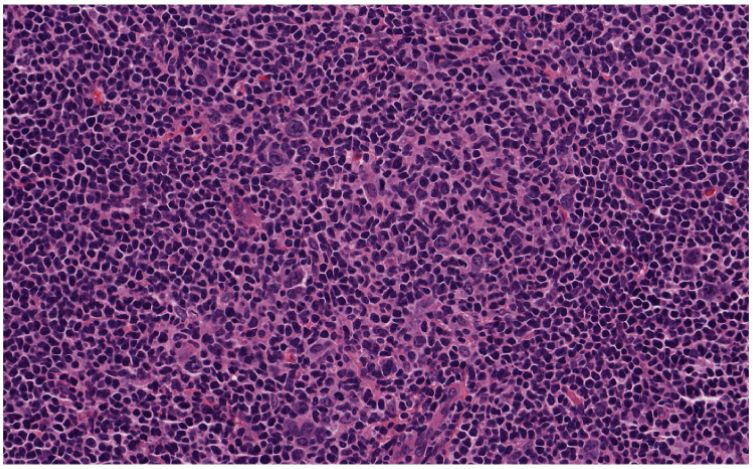
- Grade 1: 0-5 centroblasts per high-power field (HPF)
- Grade 2: 6-15 centroblasts per HPF
- Grade 3: More than 15 centroblasts per HPF

Examples for these three grades are shown in Figure 3.1.

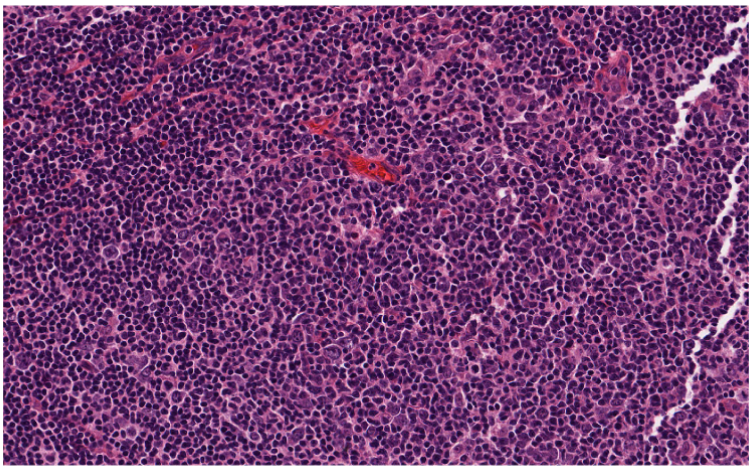
Previous studies [28] have mimicked the manual approach of pathologists, i.e., identifying the numbers of centroblasts in the sample, where the following steps are taken:

1. Segmentation of Cellular Components
2. Identifying Individual Cells
3. Centroblast Detection

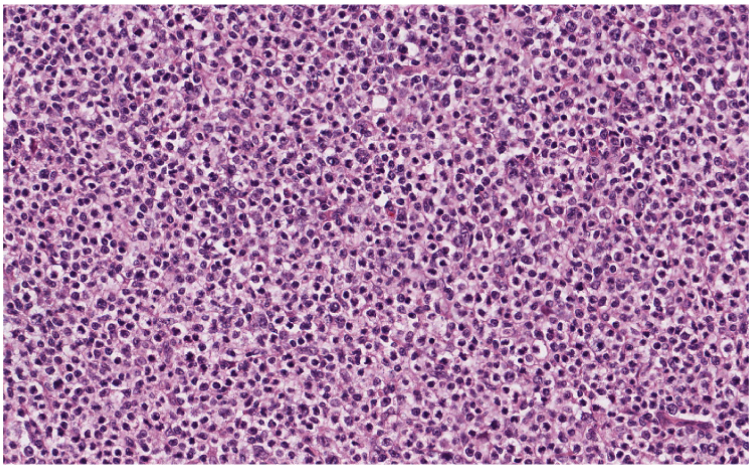
Based on the numbers of detected centroblasts, a decision on the grade of the sample can be made.



(a) FL grade 1



(b) FL grade 2



(c) FL grade 3

Figure 3.1: Examples of FL images used in this study: (a) FL grade 1, (b) FL grade 2 and (c) FL grade 3

In the present study, a different approach is chosen. The philosophy behind this is the following: Features that pathologists and molcecular biologists are interested in with respect to biomedical images are oftentimes hard to model on an image level, e.g., forms of cells may differ drastically. However, object features like this may be decomposed into several low-level components. The interplay of these low-level features can be measured by their covariance matrix, which can be used as a tool for classification. This study is a continuation of [3], where a similar classification scheme was established and which drew again on [23].

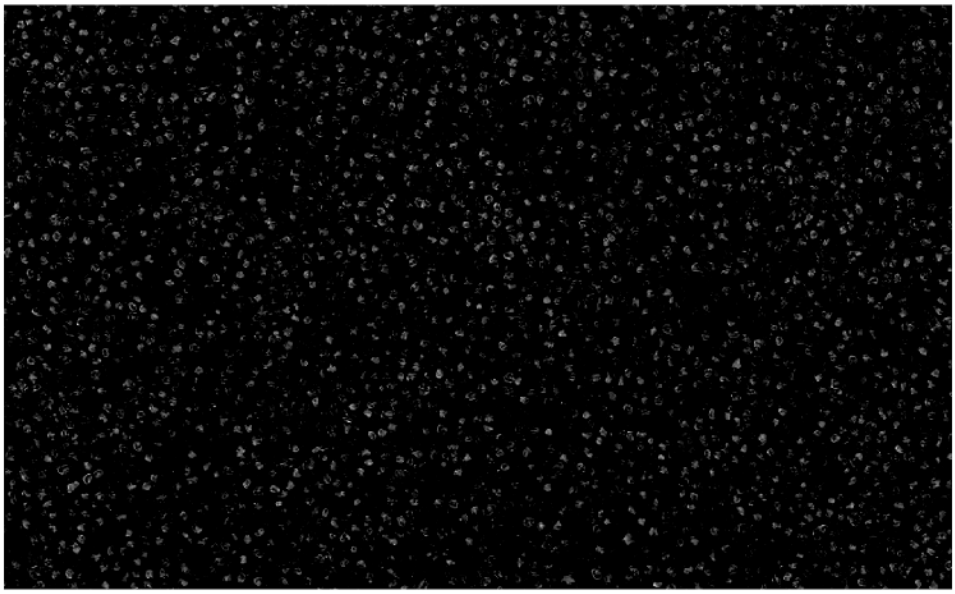
The organization of this chapter is as follows: In Section 3.1, the features used in this study are introduced. Section 3.2 deals with the adopted classification procedure. In Section 6.2 the experiments are described and discussion of the results is given.

## 3.1 Feature Extraction

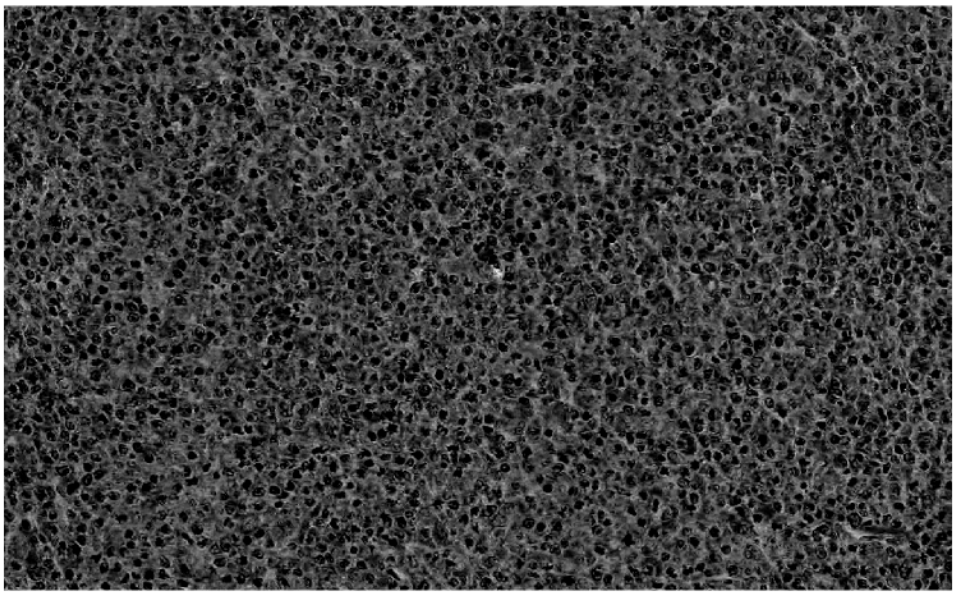
In order to achieve decent classification results, the choice of features that represents the data in a discriminative manner is of highest importance. The samples shown in the images are stained with hematoxylin and eosin (H&E) dyes, therefore displaying a specific color content. According to this observation, the following feature vector is chosen:

$$\mathbf{v}(x_i, y_j) = [L(x_i, y_j), a(x_i, y_j), b(x_i, y_j), H_p(x_i, y_j), E_p(x_i, y_i)], \tag{3.1}$$

where $L, a, b$ denote the components of the La*b*-colorspace [65] and $H_p$ and $E_p$ denote the projections of the image on the vectors $H$ and $E$, according to [66]. $H_p$ and $E_p$ therefore denote the parts of the original images that were stained with haematoxylin and eosin (H&E), respectively. Examples of these projections for the image shown in Figure 3.1 (c), can be seen in Figure 3.2. The La*b*-colorspace was chosen due to its perceptual uniformity [67].



(a) $H_p$



(b) $E_p$

Figure 3.2: Examples of of the $H_p$ and $E_p$ for the grade 3 image of Figure 3.1 (c): (a) $H_p$, (b) $E_p$

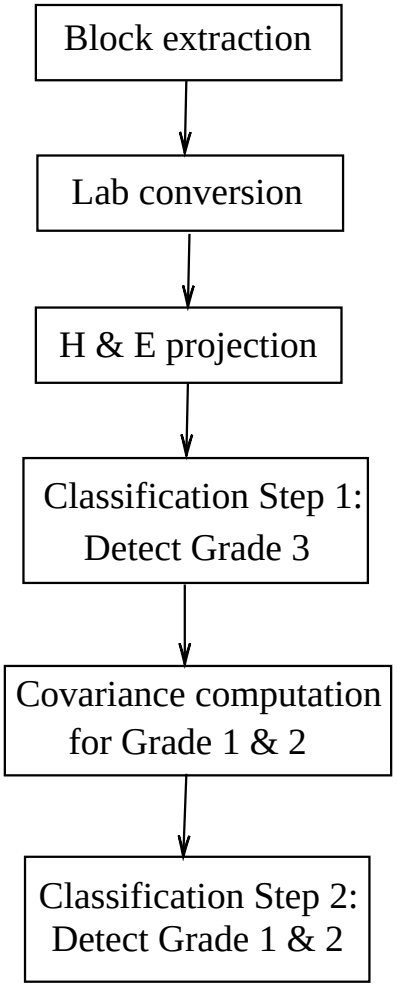



Figure 3.3: The procedure of the proposed FL classification method.

From these feature vectors a covariance matrix for a discrete 2-dimensional image region $W$ of size $M$-by-$N$ can then be obtained by using

$$\mathbf{C}(x_i, y_j) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \Big(\mathbf{v}(x_i, y_j) - \mu\Big)^T \cdot \Big(\mathbf{v}(x_i, y_j) - \mu\Big), \tag{3.2}$$

where $\mu$ is the mean vector of all feature vectors in the discrete image region. A fast computation of this procedure was introduced in [23] and used for object tracking and object recognition. The eigenvalues of this covariance matrix denote the prinicpal curvatures of the entries of the feature vector within the image region. These eigenvalues will be used as features for classification in Section 3.2.2. The whole procedure is shown in Figure 3.3. In Section 3.2.1 the features used for

classification will be the histograms of the $L$ and $H_p$ components of the feature vectors.

## 3.2 Classification

### 3.2.1 First classification stage

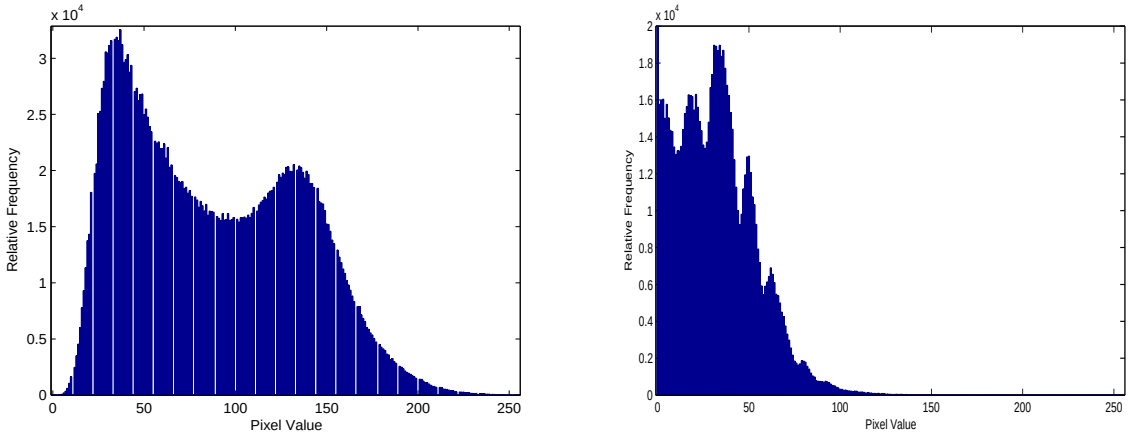
This subsection deals with the classification of grade 3 FL images. As can be seen in Figure 3.1, images from grade 1 and 2 are significantly different from grade 3 in terms of color content and brightness. This is made more clear in Figure 3.4, where the histograms of the $L$ and $H_p$ components of the example images from Figure 3.1 are shown.

Therefore, the task to be solved in this section is the following: Classify images from grade 1 and 2 into class A and images from grade 3 into class B. The procedure is the following: Histograms of the features $L$ and $H_p$ are computed for a training and test set. The histograms of the test image are compared to the average histogram of the training set using the Kullback-Leibler divergence [68]. The test image is assigned to the class where the Kullback-Leibler divergence is smallest.

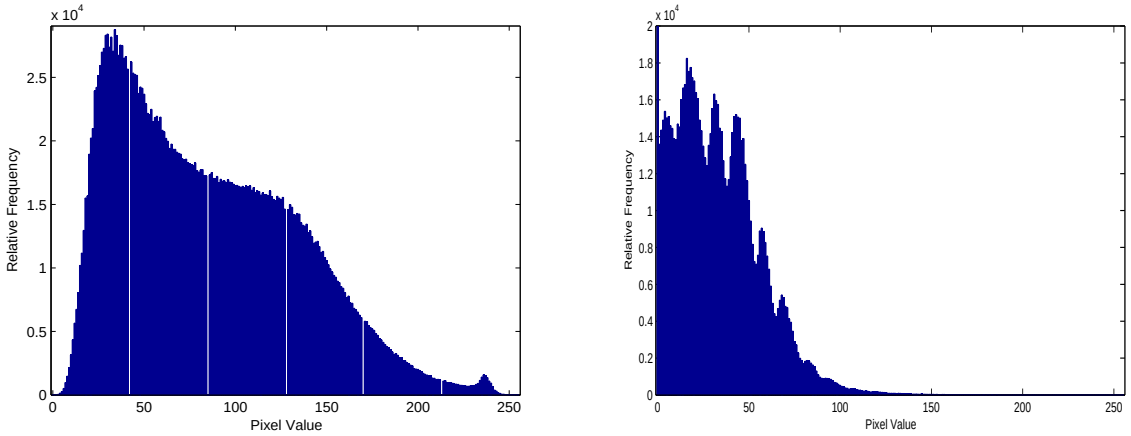
### 3.2.2 Second classification stage

In the previous section images from grade 1 and 2 were assigned to class A and images from grade 3 were assigned to class B. This section deals with the task of assigning the images in class A to grade 1 or grade 2, respectively.

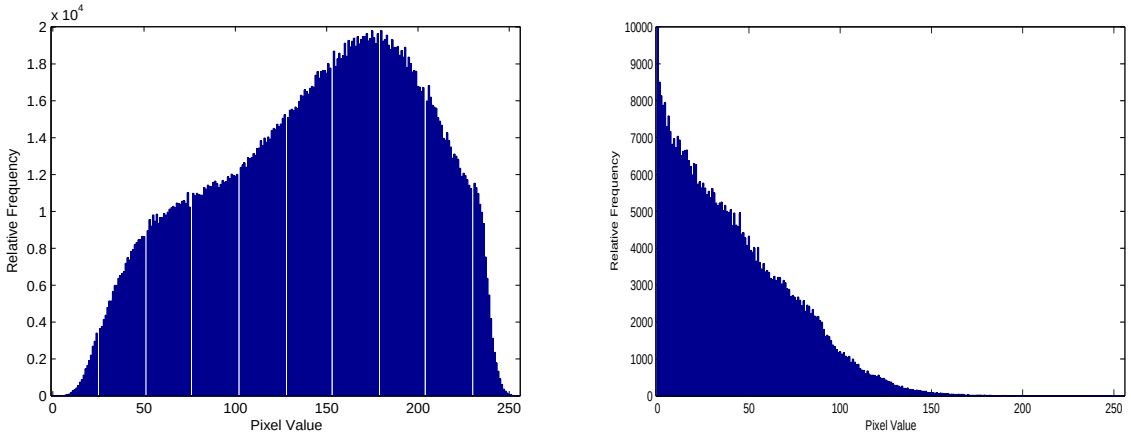
In [69] the differences between discriminative, i.e., parametric, and generative modelling approaches of posterior distributions for image classification and




(a) FL grade 1




(b) FL grade 2




(c) FL grade 3

Figure 3.4: Examples of the histograms of $L$ (right) and $H_p$ (left) used in the first step of classification: (a) FL grade 1, (b) FL grade 2, (c) FL grade 3.

object recognition were explored. It was found that generative approaches, while computationaly more costy, yield better performance than their discriminative counterparts in general. However, it is usually not straightforward to model a posterior distribution given a set of data. Computing a histogram from the data as an estimate for the posterior distribution may not work due to the limited size of the training data and the resultant unoccupied bins in the estimate of the distribution. Unlike [69] no artificial probability models are assumed in order to estimate a Bayesian model from the image data. In this chapter, a sparse representation in the transform domain of the feature parameter data is assumed. This allows us to fill the missing holes in the distribution in a smooth manner. In the well-known Bayesian estimation problem, the posterior distribution is given by:

$$p(l|\mathbf{v}) = \frac{p(\mathbf{v}|l)p(l)}{\sum_i p(\mathbf{v}|i)p(i)}, \tag{3.3}$$

where $l$ denotes the class label and $\mathbf{v}$ denotes some feature vector of the image data. In this case, the random vector $\mathbf{v}$ represents the eigenvalues of the region covariance matrix and all classes are equally likely. If prior distributions are known, one needs to compute the likelihood function in order to find the desired posterior distribution. Therefore, the rest of this section will focus on the likelihood function.

As discussed above, an estimate for the likelihood in Eq. 3.3 can be obtained easily by computing a histogram from the training data set. However, a specific $\mathbf{v}$ from the test set may not be included in the training set, therefore, calculating the likelihood and the posterior distribution for the feature vector in question may not be possible using the histogram only. The proposed method uses sparse representations of the data histograms in discrete cosine transform (DCT) domain to compute an estimate of the full likelihood function. Note that no form of the distribution, e.g. Gaussian, of the data is assumed beforehand. The only assumption is that the conditional probability density functions (PDF) has a sparse representation in either Fourier, DCT or wavelet domain. Using the sparsity of

a signal in a transform domain has been widely used in signal processing in digital waveform coding including JPEG and MPEG family of standards. In recent years, the concept of sparsity is also used to estimate a given signal from random measurements in compressive sensing (CS) [30]. The proposed method uses sparsity in DCT domain to estimate the conditional PDFs from the histogram of image features. Intuitively, one may expect likelihood functions of continuous random variables obtained from natural images as smooth functions, i.e., their representation in Fourier, DCT or wavelet domain is somewhat sparse, because natural signals and images are relatively smooth and compressible by DCT based algorithms such as JPEG. In the proposed method, the histogram $h(\mathbf{v}|l)$ of each image class is computed from the training set. Then, $h(\mathbf{v}|l)$ is transformed into Fourier space or one of its derivatives. In the experiments carried out for this study, the DCT which is the most widely used image transformation method is used:

$$g(\mathbf{Y}|l) = \mathsf{DCT}\{h(\mathbf{v}|l)\} \tag{3.4}$$

where $g(\mathbf{Y}|l)$ represents the DCT domain coefficient vector (array in multidimensional case). Size of the DCT should be selected larger than the bins of the histogram. The DCT has fast $\mathcal{O}(N \cdot logN)$ implementations.

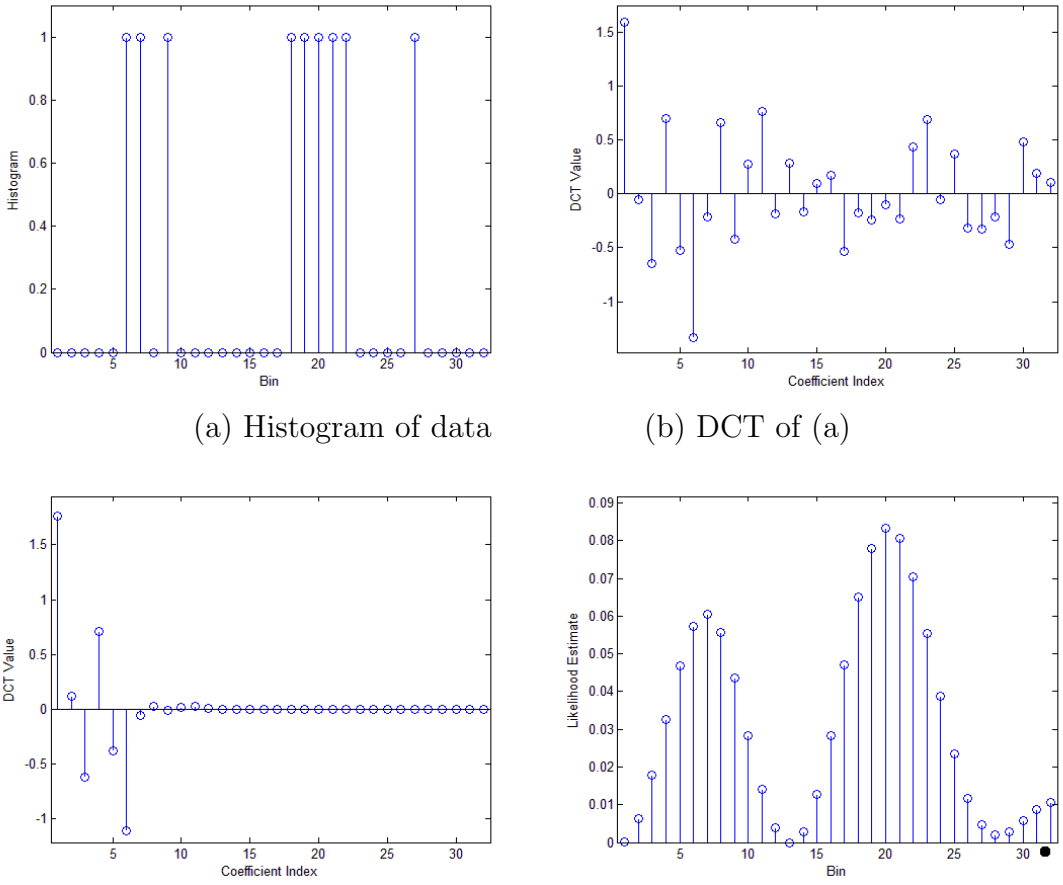
Using the sparsity assumption, only the highest $q$ percent of DCT domain coefficients are retained and the remaining DCT coefficients are set to zero. All band-limited signals are of infinite extent in the frequency domain, therefore all the gaps are filled when the reduced $\hat{g}(\mathbf{Y}|l)$ is inverse transformed to its original domain. As in natural images most of the highest $q$ percent of DCT coefficients turn out to be in the low-pass band corresponding to low indexed DCT coefficients. This $q$ percent filtered DCT coefficient set is inverse transformed to its original domain and normalized. The resultant $\hat{p}(\mathbf{v}|l)$ is an estimate of the unknown likelihood of the data. The estimated $\hat{p}(\mathbf{v}|l)$ turns out to be smooth functions because of the low-pass "filtering" in the DCT domain. After this estimation step, the posterior probability distribution on the left hand side of Eq.

3.3 can be readily computed. An example for this method is given in Figure 3.5. There is the possibility that $\hat{p}(\mathbf{v}|l)$ may contain negative values. One can then proceed to use an iterative method by setting the negative values to zero and transforming it again to DCT domain, this time retaining the $q + \#(Iterations)$ highest DCT coefficents, followed by another inverse DCT, until no negative coefficients in the resultant $\hat{p}(\mathbf{v}|l)$ are present. This iterative method will eventually converge, since it is a projection onto a convex set (POCS) [70].

In the classification step, one image is chosen as test image for each class, all other images of the respective class are assigned as training images, leading to L different classifiers, where L denotes the number of classes. Each test image is fed in a block-by-block manner to one of the L classifiers that were computed from the training set, in order to find a posterior probability for each class. For each feature a posterior PDF is computed and results are averaged. The block is assigned to the class that yields the largest posterior probability. After all blocks have been fed to the classification process, a final decision for the image is made according to the highest number of class labels for the blocks. However, only when at least 50% of all blocks in an image are assigned to one class, is the image classification counted as a success.

## 3.3 Experimental Results

For the experiments, a dataset of 90 images per grade was used. The images were of size 1313-by-2137 pixels. The size of the blocks used in Section 3.2.2 was 15-by-15. The number of blocks per image was 12126. For the experiments, a "leave-one-out" was used, i.e., the training set contained 89 images per grade and the test set contained one image per grade. The classification procedure was carried out 90 times, such that each image was at least once in the training and test sets.




(a) Histogram of data

(b) DCT of (a)

(c) Reduced DCT of (a)

(d) Inverse DCT of (c)

Figure 3.5: Depiction of the algorithm described in Section 3.2. The histogram in (a) which has empty bins is transformed into (b) using DCT. In (c) the values of the high frequency coefficients are set to zero. The inverse DCT of (c) yields an estimate for the likelihood function that has only non-empty bins, see (d). Note that in (d), bins may have very small likelihood values but larger than zero. Figure reproduced from Reference [3] with permission.

| Classification Method | Grade 1 | Grade 2 | Grade 3 |
|---|---|---|---|
| Sparsity smoothing | 98.89 | 98.89 | 100 |
| SVM | 100 | 0 | 100 |

Table 3.1: Image recognition rates (%) for all three grades and two classifiers different classifiers in the second stage: The proposed sparsity smoothing and SVM.

Two experiments were carried out. In both experiments the first stage classification from Section 3.2.1 was used. In the first experiment the second stage classification according to Section 3.2.2 was used. The feature vector used is given in Equation 3.1. In the second experiment the second stage classification was replaced with a support vector machine (SVM) using the implementation of LIBSVM [71] with a radial basis function (RBF) kernel, where $\gamma$ was chosen as 0.7 and $C$ was chosen as 1 after cross-validation.

Table 3.1 shows the image accuracy rates of the combined classification procedure. The first stage from Section 3.2.1 is identical in both cases. In the first row the results for the proposed sparsity smoothing classifier are shown, in the second row the results for the SVM are shown. For both grade 1 and 2 the proposed method misclassifies one image resulting in an accuracy of 98.8 %. For grade 3 it classifies all images correctly, i.e., 100% accuracy. The SVM shows a rather incompetitive performance not being able to classify grade 2 correctly. The weaknesses of the proposed method can be seen in Table 3.2. In this table the accuracy of the second classification step for image blocks are shown. The proposed method can on average classify less than 70% of the blocks in an image correctly. However, it does so consistently. Large image sizes are therefore vital for good results. Again, the performance the SVM is poorer.

The reason for the wrong classifications of our method in the experiments have actually been due to the first-stage classifier. One may in future investigate other features for this step, for example taking advantage of the distinct morphology of

| Classification Method | Grade 1 | Grade 2 |
|---|---|---|
| Sparsity smoothing | 67.83±3.37 | 69.64±3.93 |
| SVM | 63.92±9.44 | 19.72±0.66 |

Table 3.2: Mean and standard deviations of block recognition rates (%) for grades 1 and 2 and two different classifiers in the second stage: The proposed sparsity smoothing and SVM.

| Feature | Class A | Class B |
|---|---|---|
| $L$ | 0.0051± 0.0168 | 0.1588±0.0979 |
| $H_p$ | 0.0099± 0.0288 | 0.0275±0.0207 |

Table 3.3: Mean and standard deviations of Kullback-Leibler divergence for test images from Grade 1 to the training images of class A (grade 1 and 2) and class B (grade 3) with $L$ and $H_p$ features.

the images. In Tables 3.3 - 3.5, the average Kullback-Leibler divergences of the test set to the training set for class A and B are shown with respect to all three grades. The tables show the behavior of both features used in the first-stage classifier.

| Feature | Class A | Class B |
|---|---|---|
| $L$ | 0.0082±0.0070 | 0.1027±0.0387 |
| $H_p$ | 0.0036± 0.0024 | 0.0142±0.0062 |

Table 3.4: Mean and standard deviations of Kullback-Leibler divergence for test images from grade 2 to the training images of class A (grade 1 and 2) and class B (grade 3) with $L$ and $H_p$ features.

| Feature | Class A | Class B |
|---|---|---|
| $L$ | 0.1681± 0.0396 | 0.0147 ± 0.0115 |
| $H_p$ | 0.0188± 0.0085 | 0.0027 ± 0.0025 |

Table 3.5: Mean and standard deviations of Kullback-Leibler divergence for test images from Grade 3 to the training images of class A (grade 1 and 2) and class B (grade 3) with $L$ and $H_p$ features.

# Chapter 4

# BANDWIDTH SELECTION IN KERNEL DENSITY ESTIMATION USING FOURIER DOMAIN CONSTRAINTS

Estimating an underlying distribution from data is a widely studied problem [25]. Probability density function (PDF) estimation approaches can broadly be divided into two classes: parametric estimation and non-parametric estimation. An important branch of non-parametric estimation is the kernel-based approach, which is frequently referenced as kernel density estimation (KDE) [27]. In this approach, an estimate for the underlying density $g_X(x)$ is given by

$$\hat{g}_X(x;\sigma) = \frac{1}{N}\sum_{i=0}^{N} k_\sigma(x - v_i), \tag{4.1}$$

where $N$ is the number of data points, $v_i$ with $i = 0, 1, 2, ..., N$ denotes the observed data and $\sigma$ is called the bandwidth of the kernel $k_\sigma$, which corresponds to the standard deviation for Gaussian kernels.

The performance of KDE depends largely on the bandwidth of the kernel, which, if not chosen appropriately, can result in an over-smoothed or an under-smoothed estimate for larger than or smaller than the appropiate bandwidth, respectively. An appropiate choice for the bandwidth should decrease the mean integrated square error (MISE) [72]. In [32] several techniques for bandwidth estimation have been evaluated and it has been concluded that the most efficient method is the "plug-into-equation" approach of Sheather, *et al.* [31], [73], where the bandwidth $\sigma$ is determined by minimizing the MISE [72]. For mathematical tractability, minimization of the approximate MISE (AMISE) is carried out in [72], and the optimal $\sigma$ can be chosen accordingly. More clearly, the MISE can be written as

$$MISE\left(\hat{g}_X(x;\sigma)\right) = E\left\{\int \left(\hat{g}_X(x;\sigma) - g_X(x)\right)^2 \,\mathrm{d}x\right\}, \tag{4.2}$$

where $E\{\cdots\}$ denotes the expectation operator. Conversely, if $(g_X(x)$ has a continuous $p$-th derivative, the AMISE can be written as

$$AMISE(\sigma) = \frac{1}{N\sigma}\int k_\sigma^2(x)\,\mathrm{d}x + \sigma^{2p}\left(\frac{\mu_p k_\sigma(x)}{p!}\right)^2 \cdot \int \left(g_X^p(x)\right)^2 \,\mathrm{d}x. \tag{4.3}$$

Minimizing Eq. (4.3) with respect to $\sigma$, one arrives at the following bandwidth estimator:

$$\hat{\sigma}_{AMISE} = \left[\frac{(p!)^2 \int k_\sigma^2(x)\,\mathrm{d}x}{2p\,\mu_p^2\, k_\sigma(x)\, \int \left(g_X^p(x)\right)^2 \,\mathrm{d}x}\right]^{\frac{1}{2p+1}}. \tag{4.4}$$

The $p$-th derivative of $g_X(x)$ has to be estimated from a parametric family, since normally we don't have access to the original distribution. This is called the "pilot estimate" and relies on a-priori assumptions, which may be or may not be correct. "Pilot estimates" will be discussed further in the following paragraph.

There are also other ways of estimating the bandwidth. Reference [27], for example, discusses "leave-one-out" cross-validation (CV) as another tool. In [32],

this method was evaluated as somehow inferior to the "plug-into-equation" approaches, therefore the general opinion in the statistics community is that CV-type approaches tend to produce under-smoothed estimates of a distribution. However, Loader [33] is at odds with this broad categorisation, stating that "the comparisons between classical and plug-in approaches presented in the literature have several weaknesses. First, plug-in approaches, through the specification of tuning parameters for pilot estimates, effectively make substantial prior assumptions about the required bandwidth and will fail if this information is wrong. Second, the plug-in approaches obtain much of their information from the data through the use of higher order pilot estimates; if classical approaches are also allowed to consider higher order methods, better estimates result. Third, plug-in methods are not rescued by asymptotic analysis showing better rates of convergence; assumptions about the underlying function make the resulting estimate asymptotically inefficient, regardless of how good the bandwidth selector is."

In this chapter, we propose a method for estimating the bandwidth of the kernel by minimizing a new cost function consisting of the CV term and an $l_1$ constraint implemented in the Fourier domain. The computational cost of implementing the estimator in Eq. 4.1, is $\mathcal{O}(N^2)$ multiplications, which for large datasets may be prohibitive. A significant number of studies have been devoted to KDE, especially with the goal of reducing its computational burden [74], [73] and [75]. In [75], Eq. (4.1) is implemented in the Fourier domain via multiplication. The order of solving Eq. (4.1) in the Fourier domain is then $\mathcal{O}(Nlog(N))$ by using the fast Fourier Transform (FFT) algorithm. The proposed method takes advantage of the sparsity of the data in the Fourier domain.

In Section 4.1 we review the CV method as proposed by Silverman [27] and propose a new cost function for CV that includes an $l_1$ term. In Section 4.2, we present simulation results.

## 4.1 CV-based cost function for Bandwidth Estimation

In [27], the least-squares cross-validation was introduced as a way to find an estimate $\hat{g}_X(x;\sigma)$ that minimizes the integrated square error (ISE)

$$\begin{aligned} ISE(\hat{g}_X(x;\sigma)) &= \int (\hat{g}_X(x;\sigma) - g_X(x))^2 \, \mathrm{d}x \\ &= \int \hat{g}_X^2(x) \, \mathrm{d}x - 2\int \hat{g}_X(x;\sigma) \cdot g_X(x) \, \mathrm{d}x + \int g_X^2(x) \, \mathrm{d}x. \end{aligned} \tag{4.5}$$

Since the last term in Eq. (4.5) does not depend on the data, it is sufficient to find an estimate that minimizes the first two terms of the ISE, denoted by

$$R\left(\hat{g}_X(x;\sigma)\right) = \int \hat{g}_X^2(x) \, \mathrm{d}x - 2\int \hat{g}_X(x;\sigma) \cdot g_X(x) \, \mathrm{d}x. \tag{4.6}$$

According to [27], Eq. (4.6) can be estimated by

$$M_0(\sigma) = \int \hat{g}_X^2(x;\sigma) \, \mathrm{d}x - \frac{2}{N} \sum_{l=0, l\neq i}^{N} \hat{g}_{X-i}(v_l;\sigma), \tag{4.7}$$

where $\hat{g}_{X-i}(v_l)$ denotes the discrete density estimate constructed from all the datapoints except the $i$-th observation $v_i$.

In the CV approach, the bandwidth $\sigma$ is estimated by minimizing $M_0(\sigma)$. In the following, the problem is stated in the Fourier domain: Let $g_{\mathbf{x}}(x)$ denote the original distribution from which $N$ samples are drawn independently. Eq. (4.1) can be written as a convolution as follows:

$$\hat{g}_{\mathbf{x}}(x;\sigma) = k_\sigma(x) * \frac{1}{N} \sum_{i=0}^{N-1} \delta(x - v_i), \tag{4.8}$$

It is straight forward to see that the Fourier transform of Eq. (4.8) is

$$\hat{G}_{\mathbf{x}}(\omega;\sigma) = K_\sigma(\omega) \cdot \frac{1}{N} \sum_{i=0}^{N-1} e^{-j\omega v_i} = K_\sigma(\omega) \cdot \hat{H}(\omega), \tag{4.9}$$

where $K_\sigma$ and $\hat{H}(\omega)$ are the Fourier transforms of the kernel $k_\sigma$ and the data, respectively. Implementation of Eq. (3) is carried out using the Discrete Fourier

Transform (DFT). A discrete estimate $\hat{H}[k]$ of $\hat{H}(\omega)$ can easily be obtained by using uniform binning of the data in the interval $[-L, L]$ into $N$ intervals and computing the FFT of the binned data, which is the histogram $\hat{h}[i]$. If the kernel $k_\sigma(x)$ is chosen to be a Gaussian function with variance $\sigma$, its Fourier transform $K_\sigma(\omega)$ is again a Gaussian and can be written as

$$K_\sigma(\omega) = e^{\frac{-\sigma^2\omega^2}{2}}. \tag{4.10}$$

which is also discretized in the DFT based implementation. In the proposed approaches, $\sigma$ will be estimated as the minimizer of the $l_1$ norm of $\hat{g}_X(x;\sigma)$.

Tables 4.1 and 4.2 show that cross-validation is in fact inferior to Sheather's method when measured over our dataset. To improve the performance of cross-validation based techniques we propose to introduce an $l_1$ norm constraint in the Fourier domain.

### 4.1.1 Cross-validation using $l_1$ Norm in Fourier Domain

As mentioned above, the least-squares cross-validation tends to under-smooth its estimate of a distribution. Under-smoothing means that the Fourier transform of the estimate has large support in Fourier domain. Therefore, one would prefer estimates that are somewhat sparse in the Fourier domain. A large body of literature on sparsity exists, e.g., in [30]. Sparsity in the Fourier domain can be achieved by minimizing the $l_0$ norm of the DFT coefficients. While computing the $l_0$ norm of a signal is NP-hard, we can approximate it by the $l_1$ norm, which is far easier to compute. Taking this into account, we propose the new cost function as follows:

$$\min_{\hat{\sigma}} M_0(\sigma) + \lambda \cdot |\mathcal{F}\{\hat{g}_X(x;\sigma)\}|_1, \tag{4.11}$$

where the second term denotes the $l_1$ norm of the discrete Fourier transform of $\hat{g}_X(x;\sigma)$ as follows

$$|\mathcal{F}\{\hat{g}_X(x;\sigma)\}|_1 = \sum_{k=0}^{\frac{K}{2}-1} |G(k)|, \tag{4.12}$$

where $K$ is the DFT size and $G(k)$ denotes the DFT coefficients.

The parameter $\lambda$ in (4.11) is the linear combination parameter. Since the first term is in the sample domain and the second term is in Fourier domain they have different proportions. We want them to contribute to the overall cost function (4.11) in an approximately equal manner.

The remaining problem is now to find a suitable $\lambda$. One can carry out a greedy search to find suitable parameter values, but this would make the method computationally inefficient. One may be interested in finding an estimate for $\lambda$ from the data.

We propose to choose $\lambda$ between $\frac{1}{Nlog_2N}$ and $\frac{1}{N^2}$. The DFT coefficients are defined as

$$V[k] = \sum_{n=0}^{N-1} v[n] e^{\frac{-j2\pi kn}{N}} \tag{4.13}$$

and are much larger than sample domain coefficients in terms of amplitude. In our experiments, a $\lambda$ value of $\frac{1}{\sigma_a Nlog_2N}$, where $\sigma_a$ denotes the standard deviation of $|\mathcal{F}\{\hat{g}_X(x;\sigma)\}|$, yielded acceptable results.

### 4.1.2 Cross-validation using $l_1$ Norm and Filtered variation

The performance of the cost function defined in (4.11) can be increased by using a filter $w[n]$ as follows:

$$\min_{\hat{\sigma}} M_0(\sigma) + \lambda \cdot |\mathcal{F}\{\hat{g}_X(x;\sigma)\} \cdot W(\omega)|_1, \tag{4.14}$$

where $W(\omega)$ denotes the Fourier transform of the filter $w[n]$.One can have many choices of windows and filters. The so-called filtered variation (FV) framework was introduced in [76]. In this chapter we investigate the following filters: $w_h[n] = \{\frac{1}{2}, \frac{1}{2}\}$, $w_l[n] = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$ and $w_q[n] = \{\frac{1}{16}, \frac{1}{8}, \frac{3}{16}, \frac{1}{4}, \frac{3}{16}, \frac{1}{8}, \frac{1}{16}\}$. The first filter $w_h[n]$ is the Haar filter, the second filter $w_l[n]$ is a simple half-band filter and the third one, $w_q[n]$, has an approximate cut-off frequency of $\frac{\pi}{4}$ in the discrete time Fourier domain.

We will present experimental results in the next section. It can be seen in Tables 4.5 - 4.10 that filtering the Fourier domain data generally produces better results than using only the $l_1$ norm constraint.

## 4.2 Simulation Results

The performance of the proposed methods are evaluated by using 15 test distributions from [72]. These are mixtures of Gaussians of several flavors. Some of the original PDFs are smooth and unimodal, some of them are multimodal and have sharp peaks. $N$ random variates were independently drawn from each of the distributions. The performances of the methods under test were measured against the original test distributions under two different error criteria: Kullback-Leibler (KL) divergence and Mean Integral Square Error (MISE).

In the experiments, the influence of different choices of $N$ on the performance of the proposed method was investigated. $N$ was varied between $[2^6, 2^7, ..., 2^{12}]$. For each $N$, the experiments were again carried out 500 times for each distribution and results were averaged. Results can be seen in Tables 4.1-4.10. In these tables, the gain in dB over Sheather's method is given for different distributions and different $N$ values. Let $M_s$ denote the value of either KL divergence or MISE for Sheather's method and let $M_p$ denote the respective value for our proposed

method. Then the gain in dB can be defined as

$$G = 10 \cdot log_{10} \frac{M_s}{M_p}. \tag{4.15}$$

A positive value of $G$ in the tables suggests that our proposed method yields lower Kl-divergence or MISE than Sheather's method and is therefore preferable. We used Raykar's implementation [73] of Sheather's method. Note that both this implementation and our proposed methods are self-adjusting, i.e., the only input is the actual data and no parameter tuning is necessary. Tables 4.1 and 4.2 show the results for traditional CV that minimzes (4.6). It is clear to see that traditional CV is in most cases inferior to Sheather's method, as it tends to over-smooth the estimates of the distributions. Tables 4.3 and 4.4 show the results for the method proposed in subsection 4.1.1. Here we add the $l_1$ term to the minimzation, according to (4.11). The results instantly improve, and our method performs on average better than Sheather's method. The added $l_1$ term balances out the tendency of traditional CV to over-smooth. In Tables 4.5 - 4.10 we show the results when we introduce filtering to CV as mentioned in (4.14) from subsection 4.1.2. Again, this method performs better than CV with only $l_1$ and no filtering. These results suggest that the proposed cross-validation method including the $l_1$ term performs better than Sheather's method. However, it does not hold for all values of $N$. Our experiments suggest that our proposed methods show better performance than Sheather's method for $N \geq 128$. In this range the achieved gain is as high as approximately 2 dB. The performance can be increased by adding low-pass filters to the constarints as shown in subsection 4.1.2. In our experiments $\lambda$ was consistently chosen as $\frac{1}{\sigma_a N log_2 N}$, as explained in subsection 4.1.1.

Some examples are given for $N = 1024$ and one set of data drawn from the 15 example distributions. Our method using the $l_1$ term and $w_q[n]$ was used for these figures. Figure 4.1 shows the estimated $\sigma$ values for all distributions. In this figure, traditional CV's tendency of over-smoothing is hinted at, since

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -20.78 | -25.48 | -29.48 | -31.21 | -30.05 | -26.72 | -23.75 |
| 2 | -17.84 | -23.40 | -27.46 | -29.07 | -27.78 | -24.71 | -21.72 |
| 3 | -1.95 | -6.93 | -11.52 | -14.22 | -13.28 | -11.01 | -8.11 |
| 4 | -9.07 | -13.37 | -17.14 | -18.22 | -16.52 | -13.10 | -9.81 |
| 5 | -2.68 | -5.69 | -9.22 | -11.67 | -11.96 | -9.97 | -7.01 |
| 6 | -18.77 | -24.22 | -27.77 | -30.89 | -28.11 | -25.23 | -22.51 |
| 7 | -19.92 | -25.26 | -28.08 | -28.58 | -26.61 | -24.38 | -20.52 |
| 8 | -17.27 | -21.98 | -25.89 | -28.17 | -25.94 | -22.93 | -19.43 |
| 9 | -17.83 | -23.76 | -26.69 | -28.98 | -25.87 | -23.86 | -20.67 |
| 10 | -8.34 | -10.50 | -11.15 | -10.49 | -14.85 | -13.14 | -9.84 |
| 11 | -17.08 | -24.78 | -26.65 | -28.34 | -24.28 | -19.84 | -14.67 |
| 12 | -7.84 | -12.33 | -14.04 | -14.80 | -10.39 | -5.34 | -0.04 |
| 13 | -16.73 | -22.98 | -24.20 | -25.32 | -20.02 | -14.79 | -10.09 |
| 14 | 11.08 | 7.40 | 4.65 | 6.97 | 8.10 | 13.50 | 16.47 |
| 15 | 5.96 | -1.15 | -2.60 | 0.96 | 2.31 | 9.25 | 13.98 |
| Average | -10.60 | -15.63 | -18.48 | -19.47 | -17.68 | -14.15 | -10.51 |

Table 4.1: KL divergence gain in dB over Sheather's method for traditional cross-validation

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -23.90 | -29.18 | -33.84 | -35.76 | -33.90 | -29.85 | -26.03 |
| 2 | -20.55 | -26.75 | -31.18 | -33.29 | -30.84 | -27.07 | -23.76 |
| 3 | -0.36 | -6.64 | -10.97 | -13.50 | -11.16 | -8.36 | -4.98 |
| 4 | -1.79 | -8.05 | -13.23 | -16.25 | -14.09 | -10.89 | -7.48 |
| 5 | 1.54 | -4.66 | -11.39 | -15.43 | -14.11 | -10.17 | -6.41 |
| 6 | -24.03 | -28.59 | -32.40 | -34.96 | -31.17 | -27.96 | -24.70 |
| 7 | -23.26 | -28.58 | -31.76 | -32.22 | -29.24 | -26.61 | -22.49 |
| 8 | -20.82 | -25.36 | -29.19 | -31.79 | -28.29 | -24.64 | -20.53 |
| 9 | -23.47 | -28.15 | -30.46 | -32.33 | -28.22 | -25.54 | -21.82 |
| 10 | -5.24 | -6.75 | -7.26 | -6.63 | -12.79 | -11.69 | -8.34 |
| 11 | -19.56 | -29.04 | -28.56 | -29.80 | -23.54 | -17.26 | -10.70 |
| 12 | -10.16 | -14.18 | -15.79 | -16.85 | -12.50 | -7.53 | -2.18 |
| 13 | -18.17 | -24.27 | -24.04 | -24.21 | -16.96 | -10.42 | -4.68 |
| 14 | -6.84 | -9.27 | -12.33 | -9.47 | -6.81 | 0.06 | 4.53 |
| 15 | -9.95 | -11.60 | -13.78 | -9.08 | -5.60 | 2.42 | 7.77 |
| Average | -13.77 | -18.74 | -21.75 | -22.77 | -19.95 | -15.70 | -11.45 |

Table 4.2: MISE gain in dB over Sheather's method for traditional cross-validation.

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -20.40 | -0.21 | -0.80 | -1.48 | -1.73 | -1.85 | -2.16 |
| 2 | -17.35 | -0.66 | -1.21 | -1.47 | -1.73 | -1.61 | -1.90 |
| 3 | -1.03 | 2.83 | 2.66 | 2.01 | 1.18 | 0.52 | 0.08 |
| 4 | -8.79 | -0.89 | -1.07 | -1.03 | -0.90 | -0.69 | -0.45 |
| 5 | -2.60 | -0.20 | -0.57 | -0.43 | -0.14 | 0.04 | 0.28 |
| 6 | -18.34 | 0.96 | 0.63 | 0.06 | -0.45 | -0.52 | -0.73 |
| 7 | -19.12 | -0.32 | 0.07 | -0.02 | -0.27 | -0.50 | -0.57 |
| 8 | -16.79 | 0.44 | 0.14 | -0.12 | -0.43 | -0.64 | -0.65 |
| 9 | -17.38 | 0.49 | 0.94 | 0.48 | 0.20 | -0.25 | -0.42 |
| 10 | -8.04 | 0.99 | 4.99 | 6.85 | 0.96 | -0.25 | -0.29 |
| 11 | -16.64 | 0.77 | 0.42 | 0.05 | -0.32 | -0.35 | -0.41 |
| 12 | -7.23 | 1.89 | 2.53 | 2.03 | 2.26 | 3.27 | 4.28 |
| 13 | -16.29 | 0.46 | 0.24 | -0.09 | 0.34 | 1.21 | 2.19 |
| 14 | 11.68 | 1.30 | 4.28 | 6.91 | 9.00 | 11.02 | 12.51 |
| 15 | 6.57 | 0.70 | 2.17 | 7.85 | 11.61 | 14.54 | 16.72 |
| Average | -10.12 | 0.57 | 1.03 | 1.44 | 1.30 | 1.60 | 1.90 |

Table 4.3: KL divergence gain in dB over Sheather's method for proposed cross-validation with $l_1$ term.

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -23.68 | -0.54 | -1.31 | -2.13 | -2.32 | -2.53 | -2.77 |
| 2 | -20.25 | -0.88 | -1.45 | -1.57 | -1.82 | -1.54 | -2.03 |
| 3 | 0.19 | 3.09 | 3.68 | 3.78 | 3.26 | 2.78 | 1.95 |
| 4 | -1.64 | 1.76 | 1.66 | 1.14 | 0.83 | 0.46 | 0.24 |
| 5 | 1.56 | -0.26 | -0.57 | -0.50 | -0.28 | -0.24 | -0.26 |
| 6 | -23.79 | 0.39 | -0.09 | -0.62 | -1.05 | -1.03 | -1.21 |
| 7 | -22.78 | -0.20 | -0.22 | -0.47 | -0.79 | -0.96 | -1.01 |
| 8 | -20.54 | 0.26 | -0.03 | -0.09 | -0.27 | -0.48 | -0.40 |
| 9 | -23.21 | 0.22 | 0.31 | -0.01 | -0.02 | -0.35 | -0.43 |
| 10 | -5.08 | 1.99 | 9.33 | 12.53 | 3.83 | 1.25 | 0.66 |
| 11 | -19.32 | 0.27 | -0.11 | -0.34 | -0.52 | -0.23 | -0.12 |
| 12 | -9.81 | 1.24 | 3.04 | 3.22 | 3.33 | 3.98 | 4.56 |
| 13 | -17.92 | 0.19 | 0.03 | 0.22 | 1.63 | 3.54 | 5.06 |
| 14 | -6.32 | 0.65 | 2.15 | 3.34 | 4.29 | 5.42 | 6.71 |
| 15 | -9.48 | 0.24 | 0.63 | 3.89 | 7.52 | 10.62 | 12.95 |
| Average | -13.47 | 0.56 | 1.14 | 1.49 | 1.18 | 1.38 | 1.59 |

Table 4.4: MISE gain in dB over Sheather's method for proposed cross-validation with $l_1$ term.

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -20.54 | -0.21 | -0.81 | -1.52 | -1.74 | -1.86 | -2.17 |
| 2 | -17.55 | -0.67 | -1.22 | -1.49 | -1.75 | -1.61 | -1.92 |
| 3 | -1.44 | 2.85 | 2.65 | 2.00 | 1.17 | 0.51 | 0.06 |
| 4 | -8.88 | -0.92 | -1.10 | -1.07 | -0.93 | -0.73 | -0.47 |
| 5 | -2.62 | -0.21 | -0.59 | -0.46 | -0.16 | 0.02 | 0.27 |
| 6 | -18.51 | 0.96 | 0.63 | 0.05 | -0.46 | -0.53 | -0.74 |
| 7 | -19.46 | -0.32 | 0.07 | -0.03 | -0.28 | -0.51 | -0.58 |
| 8 | -16.98 | 0.44 | 0.13 | -0.13 | -0.45 | -0.65 | -0.66 |
| 9 | -17.55 | 0.50 | 0.93 | 0.48 | 0.20 | -0.25 | -0.43 |
| 10 | -8.15 | 0.99 | 4.99 | 6.83 | 0.94 | -0.26 | -0.31 |
| 11 | -16.81 | 0.77 | 0.42 | 0.05 | -0.33 | -0.36 | -0.41 |
| 12 | -7.47 | 1.90 | 2.53 | 2.03 | 2.27 | 3.29 | 4.32 |
| 13 | -16.46 | 0.46 | 0.24 | -0.10 | 0.33 | 1.19 | 2.18 |
| 14 | 11.45 | 1.34 | 4.35 | 7.05 | 9.17 | 11.26 | 12.69 |
| 15 | 6.33 | 0.73 | 2.24 | 8.05 | 11.74 | 14.65 | 16.82 |
| Average | -10.31 | 0.57 | 1.03 | 1.45 | 1.32 | 1.61 | 1.91 |

Table 4.5: KL divergence gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_h[n]$ was used.

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -23.77 | -0.55 | -1.32 | -2.18 | -2.33 | -2.55 | -2.78 |
| 2 | -20.39 | -0.90 | -1.47 | -1.59 | -1.84 | -1.55 | -2.05 |
| 3 | -0.08 | 3.12 | 3.69 | 3.80 | 3.28 | 2.79 | 1.96 |
| 4 | -1.70 | 1.77 | 1.67 | 1.14 | 0.83 | 0.46 | 0.25 |
| 5 | 1.55 | -0.27 | -0.59 | -0.52 | -0.28 | -0.24 | -0.26 |
| 6 | -23.89 | 0.38 | -0.09 | -0.63 | -1.06 | -1.04 | -1.22 |
| 7 | -23.00 | -0.20 | -0.23 | -0.48 | -0.80 | -0.98 | -1.03 |
| 8 | -20.66 | 0.25 | -0.04 | -0.10 | -0.28 | -0.49 | -0.40 |
| 9 | -23.32 | 0.22 | 0.30 | -0.02 | -0.03 | -0.36 | -0.44 |
| 10 | -5.14 | 2.01 | 9.36 | 12.54 | 3.84 | 1.25 | 0.67 |
| 11 | -19.42 | 0.27 | -0.11 | -0.35 | -0.53 | -0.23 | -0.12 |
| 12 | -9.97 | 1.24 | 3.05 | 3.23 | 3.34 | 4.00 | 4.59 |
| 13 | -18.03 | 0.19 | 0.03 | 0.21 | 1.64 | 3.55 | 5.07 |
| 14 | -6.56 | 0.67 | 2.17 | 3.38 | 4.35 | 5.50 | 6.79 |
| 15 | -9.69 | 0.25 | 0.65 | 3.99 | 7.59 | 10.67 | 13.00 |
| Average | -13.60 | 0.56 | 1.14 | 1.50 | 1.18 | 1.39 | 1.60 |

Table 4.6: MISE gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_h[n]$ was used.

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -20.56 | -0.22 | -0.81 | -1.53 | -1.74 | -1.87 | -2.17 |
| 2 | -17.58 | -0.68 | -1.23 | -1.50 | -1.76 | -1.62 | -1.92 |
| 3 | -1.50 | 2.86 | 2.65 | 1.98 | 1.15 | 0.50 | 0.05 |
| 4 | -8.90 | -0.94 | -1.14 | -1.10 | -0.96 | -0.75 | -0.48 |
| 5 | -2.62 | -0.22 | -0.61 | -0.47 | -0.18 | 0.01 | 0.26 |
| 6 | -18.53 | 0.97 | 0.63 | 0.05 | -0.46 | -0.54 | -0.75 |
| 7 | -19.50 | -0.32 | 0.07 | -0.03 | -0.30 | -0.52 | -0.59 |
| 8 | -17.01 | 0.44 | 0.13 | -0.14 | -0.45 | -0.66 | -0.66 |
| 9 | -17.57 | 0.50 | 0.93 | 0.47 | 0.19 | -0.26 | -0.43 |
| 10 | -8.16 | 0.99 | 4.99 | 6.80 | 0.94 | -0.28 | -0.31 |
| 11 | -16.83 | 0.76 | 0.42 | 0.04 | -0.34 | -0.36 | -0.42 |
| 12 | -7.51 | 1.90 | 2.53 | 2.03 | 2.29 | 3.31 | 4.34 |
| 13 | -16.48 | 0.46 | 0.24 | -0.10 | 0.33 | 1.18 | 2.18 |
| 14 | 11.42 | 1.38 | 4.43 | 7.17 | 9.29 | 11.39 | 12.79 |
| 15 | 6.30 | 0.75 | 2.30 | 8.25 | 11.82 | 14.70 | 16.85 |
| Average | -10.34 | 0.58 | 1.03 | 1.46 | 1.32 | 1.62 | 1.92 |

Table 4.7: KL divergence gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_l[n]$ was used.

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -23.78 | -0.56 | -1.33 | -2.20 | -2.34 | -2.56 | -2.79 |
| 2 | -20.40 | -0.91 | -1.48 | -1.61 | -1.85 | -1.57 | -2.06 |
| 3 | -0.11 | 3.14 | 3.71 | 3.80 | 3.29 | 2.80 | 1.97 |
| 4 | -1.71 | 1.78 | 1.68 | 1.14 | 0.83 | 0.47 | 0.25 |
| 5 | 1.55 | -0.28 | -0.61 | -0.52 | -0.28 | -0.24 | -0.26 |
| 6 | -23.90 | 0.38 | -0.10 | -0.63 | -1.06 | -1.05 | -1.22 |
| 7 | -23.03 | -0.20 | -0.23 | -0.49 | -0.82 | -0.99 | -1.03 |
| 8 | -20.68 | 0.25 | -0.04 | -0.11 | -0.28 | -0.50 | -0.41 |
| 9 | -23.33 | 0.22 | 0.29 | -0.03 | -0.04 | -0.37 | -0.44 |
| 10 | -5.15 | 2.03 | 9.39 | 12.55 | 3.85 | 1.26 | 0.67 |
| 11 | -19.43 | 0.24 | -0.12 | -0.35 | -0.54 | -0.23 | -0.13 |
| 12 | -9.99 | 1.25 | 3.05 | 3.24 | 3.35 | 4.01 | 4.61 |
| 13 | -18.04 | 0.19 | 0.02 | 0.21 | 1.64 | 3.54 | 5.07 |
| 14 | -6.60 | 0.69 | 2.20 | 3.42 | 4.38 | 5.55 | 6.83 |
| 15 | -9.72 | 0.26 | 0.66 | 4.09 | 7.62 | 10.69 | 13.02 |
| Average | -13.62 | 0.56 | 1.14 | 1.50 | 1.18 | 1.39 | 1.61 |

Table 4.8: MISE gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_l[n]$ was used.

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -20.57 | -0.25 | -0.86 | -1.63 | -1.81 | -1.95 | -2.24 |
| 2 | -17.59 | -0.76 | -1.32 | -1.65 | -1.89 | -1.71 | -2.00 |
| 3 | -1.52 | 2.96 | 2.62 | 1.93 | 1.11 | 0.46 | 0.03 |
| 4 | -8.90 | -1.13 | -1.37 | -1.27 | -1.07 | -0.82 | -0.52 |
| 5 | -2.62 | -0.27 | -0.69 | -0.55 | -0.22 | -0.02 | 0.25 |
| 6 | -18.54 | 0.97 | 0.61 | 0.00 | -0.53 | -0.64 | -0.79 |
| 7 | -19.53 | -0.29 | 0.05 | -0.07 | -0.36 | -0.59 | -0.64 |
| 8 | -17.02 | 0.43 | 0.10 | -0.20 | -0.51 | -0.72 | -0.71 |
| 9 | -17.59 | 0.52 | 0.93 | 0.41 | 0.13 | -0.35 | -0.48 |
| 10 | -8.17 | 1.06 | 4.93 | 6.71 | 0.85 | -0.34 | -0.35 |
| 11 | -16.84 | 0.76 | 0.40 | 0.00 | -0.40 | -0.43 | -0.46 |
| 12 | -7.53 | 1.94 | 2.54 | 2.06 | 2.37 | 3.41 | 4.46 |
| 13 | -16.49 | 0.46 | 0.21 | -0.17 | 0.24 | 1.10 | 2.13 |
| 14 | 11.40 | 1.68 | 4.99 | 7.94 | 10.07 | 11.95 | 13.03 |
| 15 | 6.28 | 0.96 | 3.00 | 9.25 | 12.27 | 14.95 | 17.00 |
| Average | -10.35 | 0.60 | 1.08 | 1.52 | 1.35 | 1.62 | 1.91 |

Table 4.9: KL divergence gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_q[n]$ was used.

| Distribution Number | N=64 | N=128 | N=256 | N=512 | N=1024 | N=2048 | N=4096 |
|---|---|---|---|---|---|---|---|
| 1 | -23.79 | -0.61 | -1.40 | -2.34 | -2.43 | -2.67 | -2.87 |
| 2 | -20.41 | -1.05 | -1.61 | -1.81 | -2.03 | -1.67 | -2.16 |
| 3 | -0.13 | 3.31 | 3.80 | 3.88 | 3.36 | 2.84 | 2.00 |
| 4 | -1.71 | 1.85 | 1.70 | 1.17 | 0.87 | 0.49 | 0.28 |
| 5 | 1.55 | -0.32 | -0.68 | -0.56 | -0.28 | -0.24 | -0.26 |
| 6 | -23.91 | 0.35 | -0.15 | -0.71 | -1.16 | -1.17 | -1.28 |
| 7 | -23.05 | -0.21 | -0.28 | -0.56 | -0.92 | -1.09 | -1.10 |
| 8 | -20.69 | 0.24 | -0.09 | -0.19 | -0.34 | -0.56 | -0.47 |
| 9 | -23.34 | 0.20 | 0.27 | -0.11 | -0.12 | -0.48 | -0.50 |
| 10 | -5.16 | 2.30 | 9.56 | 12.62 | 3.89 | 1.29 | 0.69 |
| 11 | -19.44 | 0.21 | -0.16 | -0.41 | -0.59 | -0.28 | -0.15 |
| 12 | -10.00 | 1.29 | 3.09 | 3.29 | 3.42 | 4.08 | 4.69 |
| 13 | -18.05 | 0.17 | 0.00 | 0.19 | 1.64 | 3.56 | 5.09 |
| 14 | -6.61 | 0.82 | 2.38 | 3.61 | 4.59 | 5.74 | 6.93 |
| 15 | -9.74 | 0.31 | 0.82 | 4.55 | 7.83 | 10.81 | 13.10 |
| Average | -13.63 | 0.59 | 1.15 | 1.51 | 1.18 | 1.38 | 1.60 |

Table 4.10: MISE gain in dB over Sheather's method for proposed cross-validation with $l_1$ term. $w_q[n]$ was used.

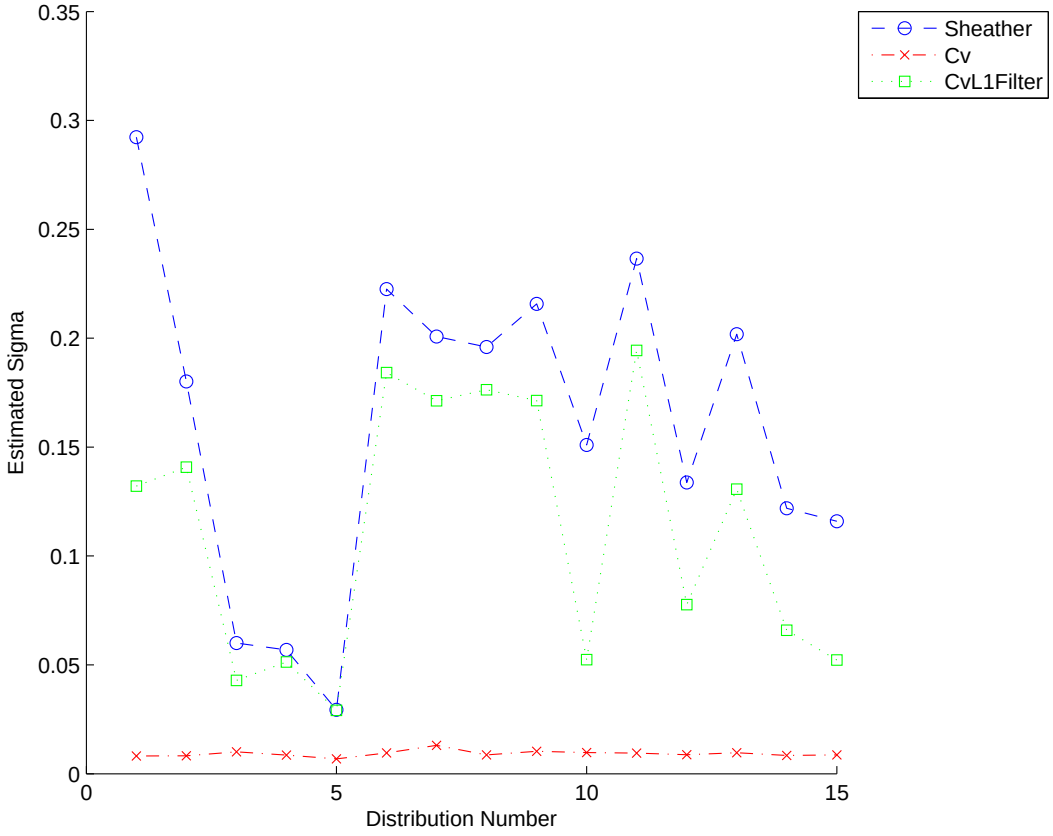
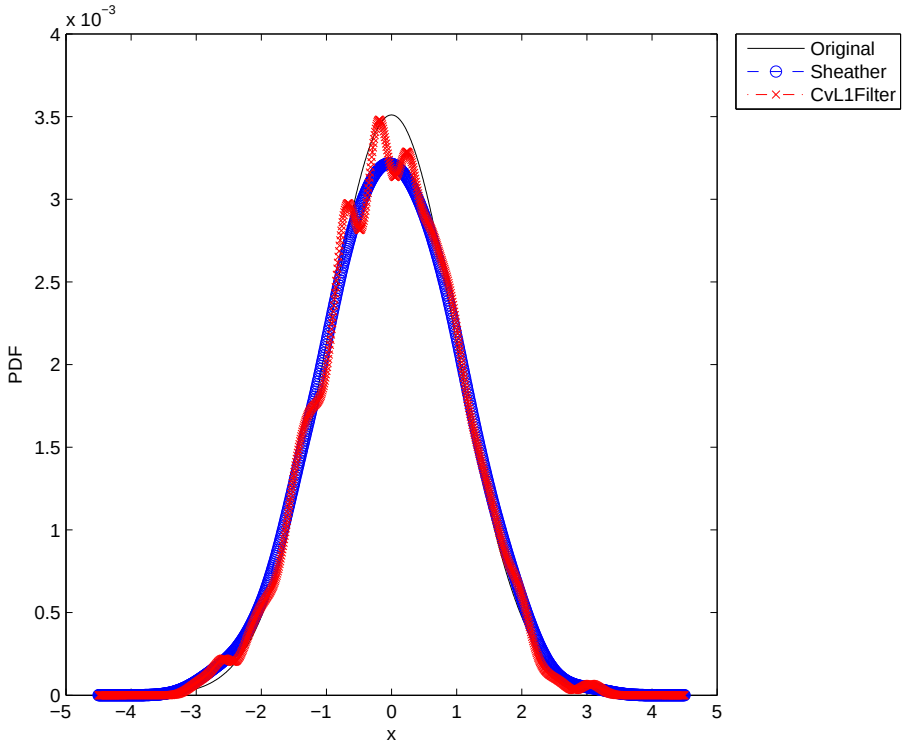
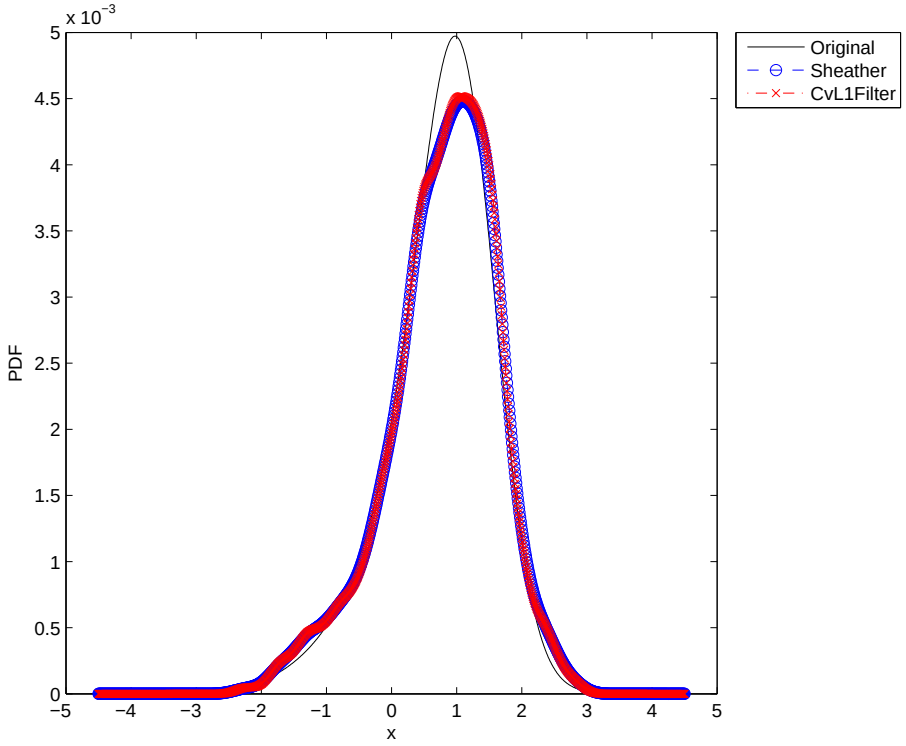



Figure 4.1: Estimated $\sigma$ for Sheather's method and the proposed method.

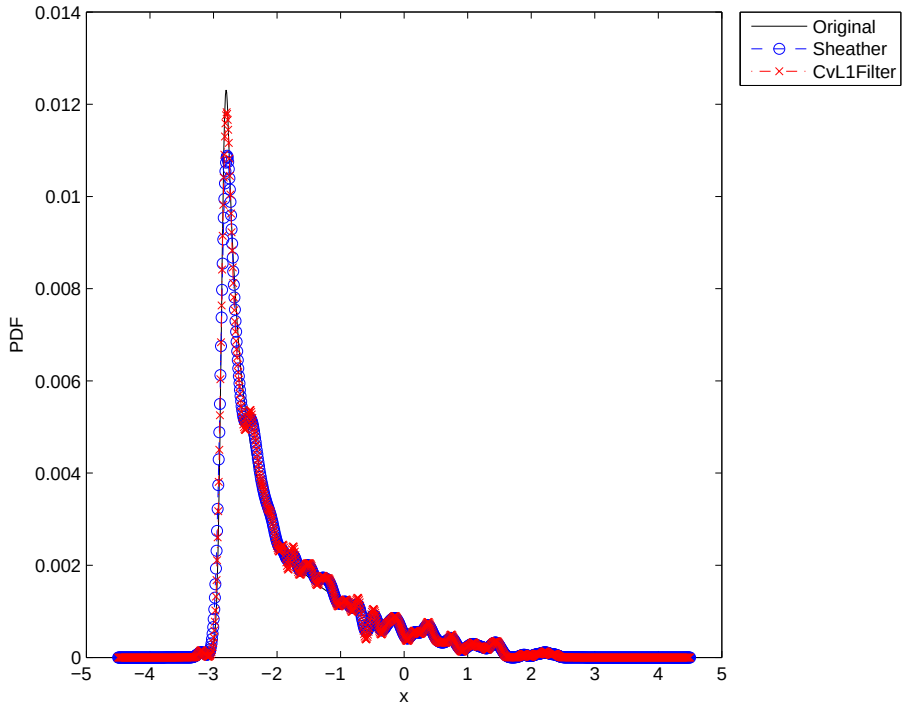
the estimated $\sigma$ is much lower than the corresponding bandwidths of Sheather's method or our proposed method. Our method's bandwidths are consistently smaller than Sheather's method, resulting in more detail in the resultant estimates. Figure 4.2 shows the density estimates of the proposed methods for three example distributions. Densities 10 and 15 shown in Figures 4.2 (j) and (o), respectivey, are multi-modal. It is easy to see that for these multimodal distributions, the proposed method performs better than Sheather's method, since they are able to pick up the high frequency content of the distribution. However, for a distribution that includes sharp spikes like distribution 11 shown in Figure 4.2 (k), our proposed method performs comparable to Sheather's method. In the next chapter we will apply the proposed bandwidth estimator to the problem of microscopic image thresholding.



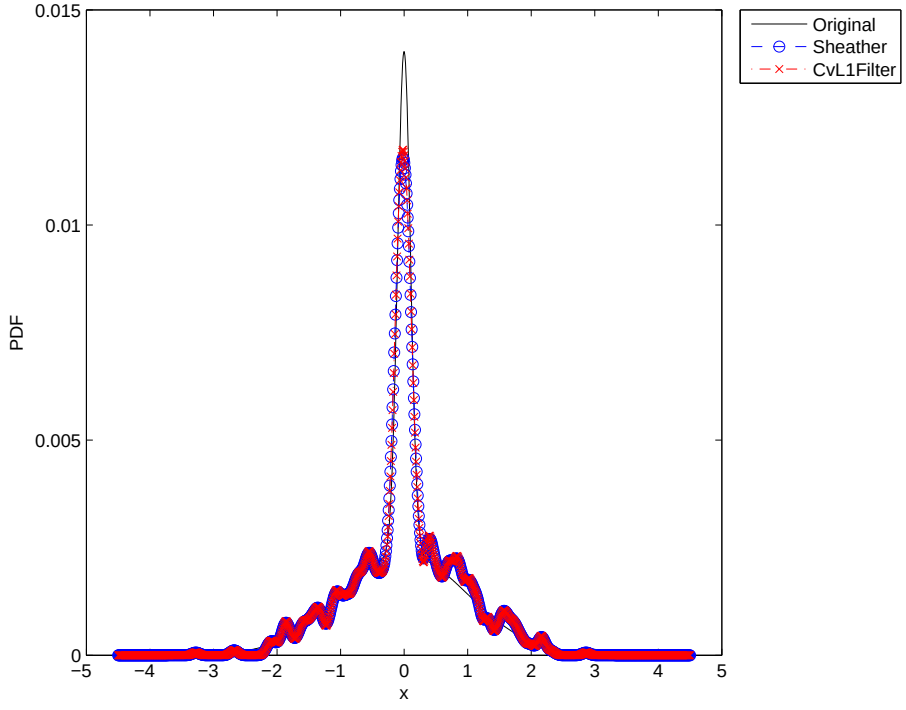

(a) Distribution 1



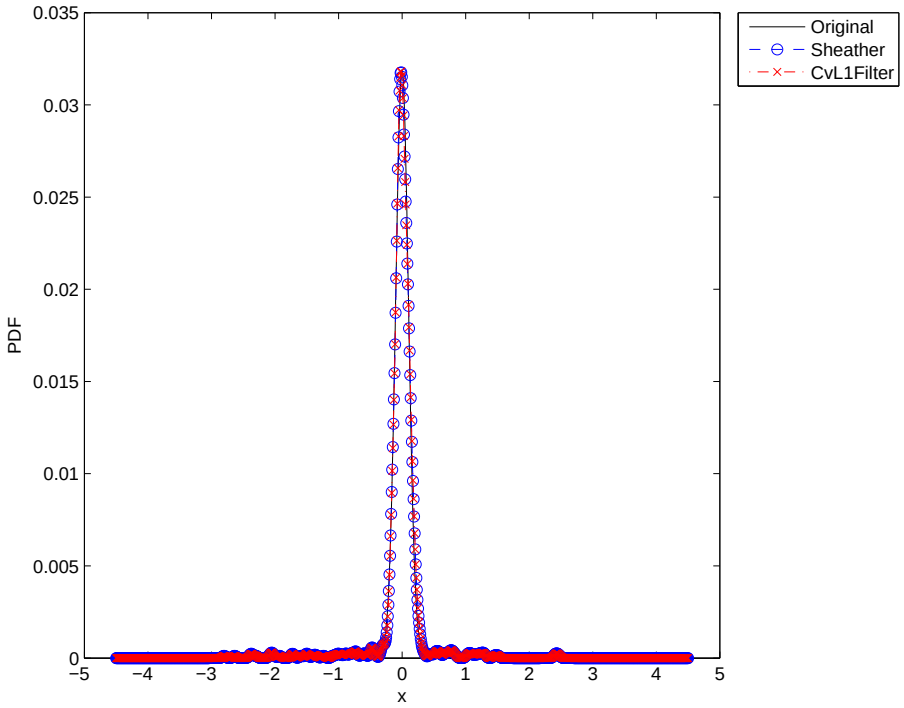

(b) Distribution 2



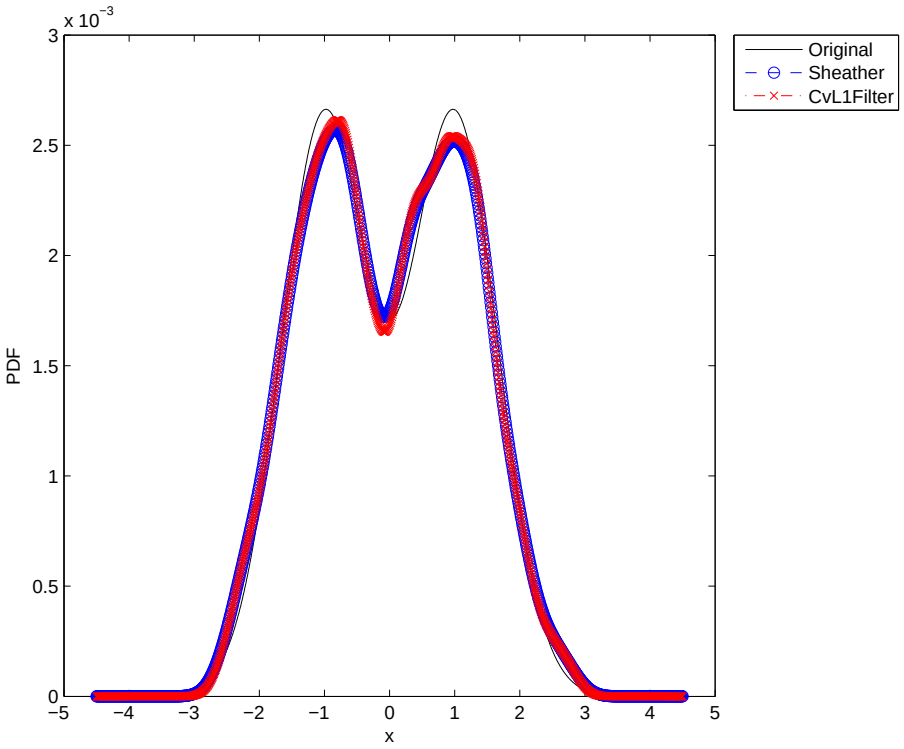

(c) Distribution 3



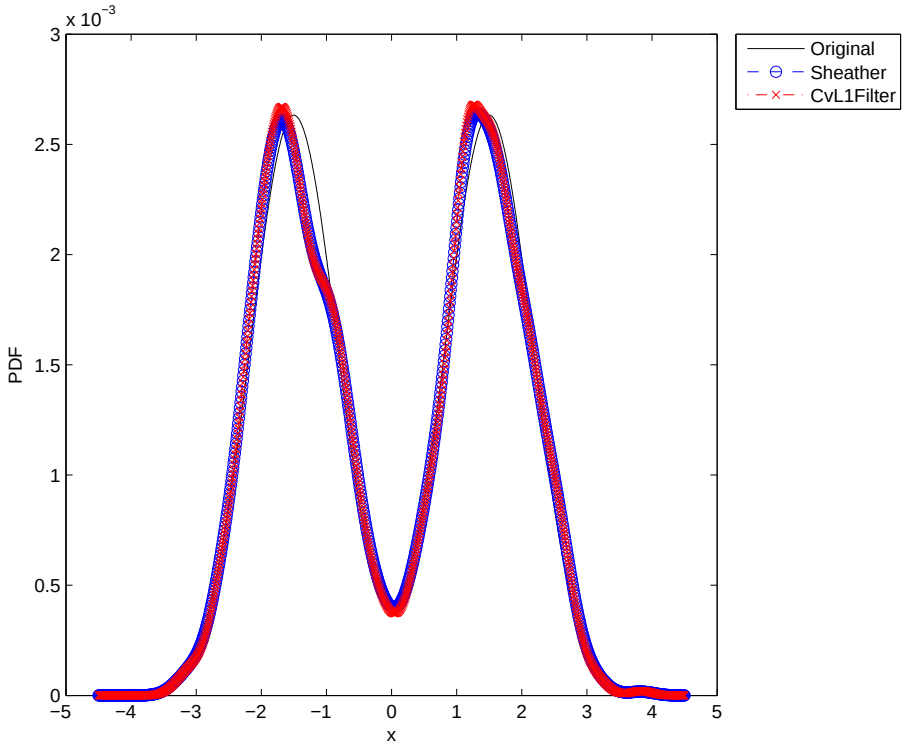

(d) Distribution 4



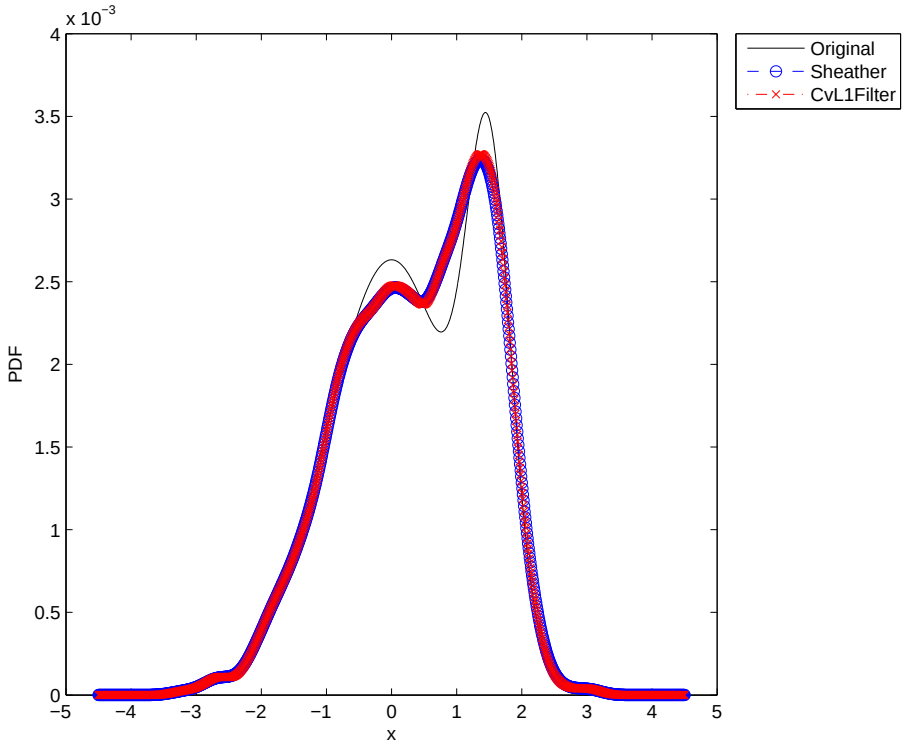

(e) Distribution 5



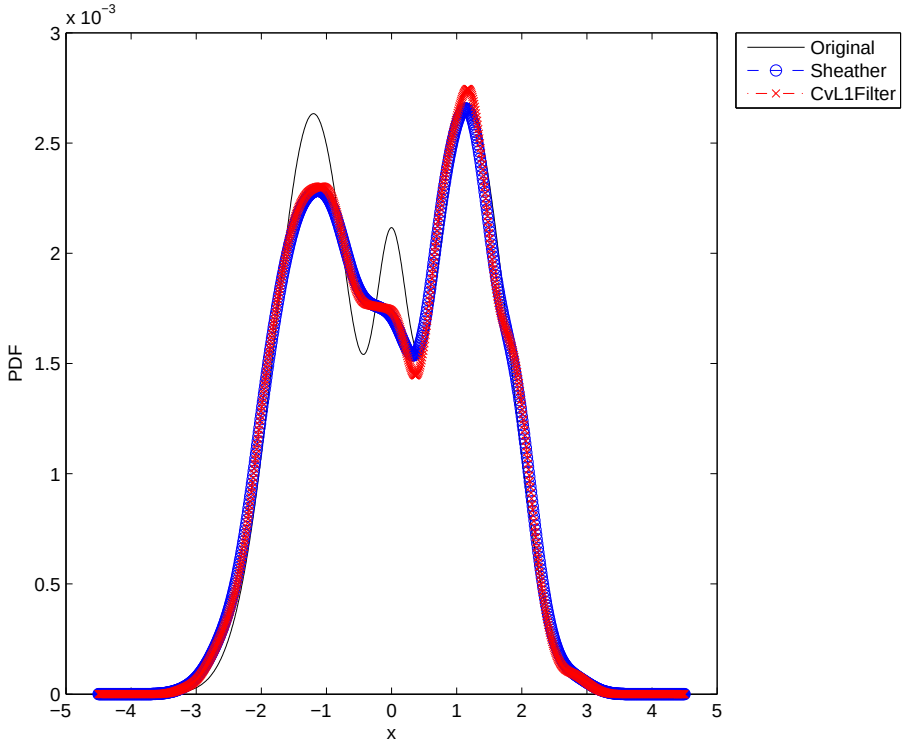

(f) Distribution 6



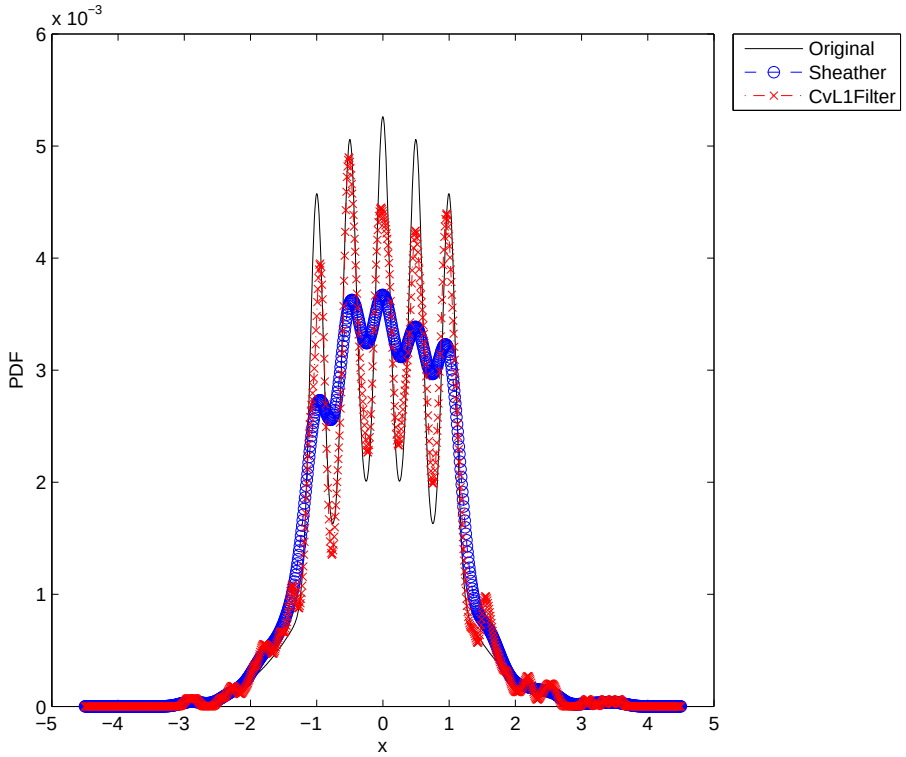

(g) Distribution 7



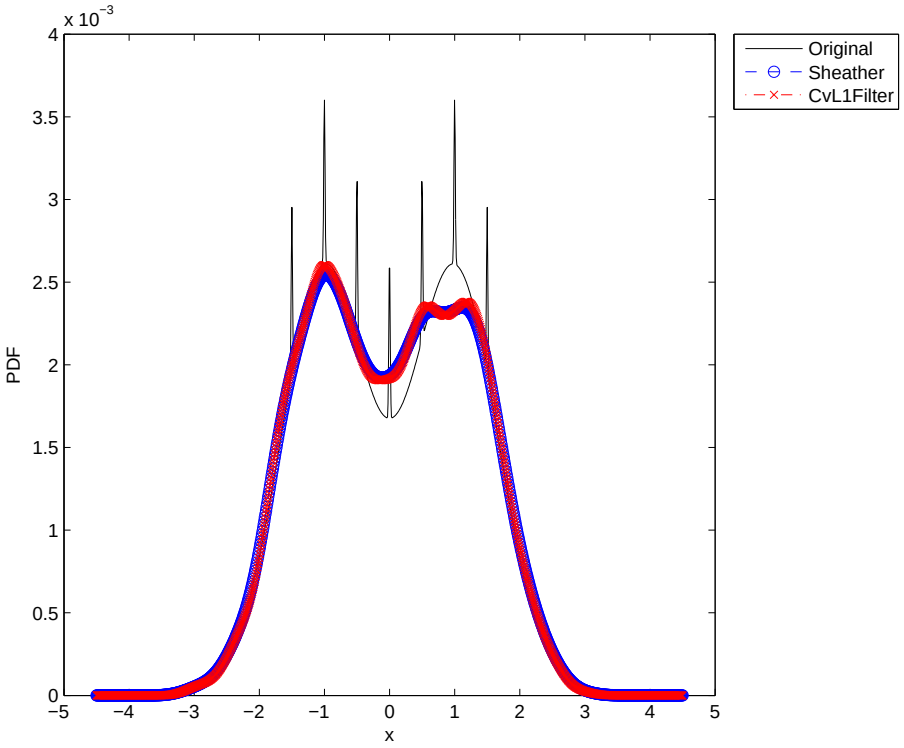

(h) Distribution 8



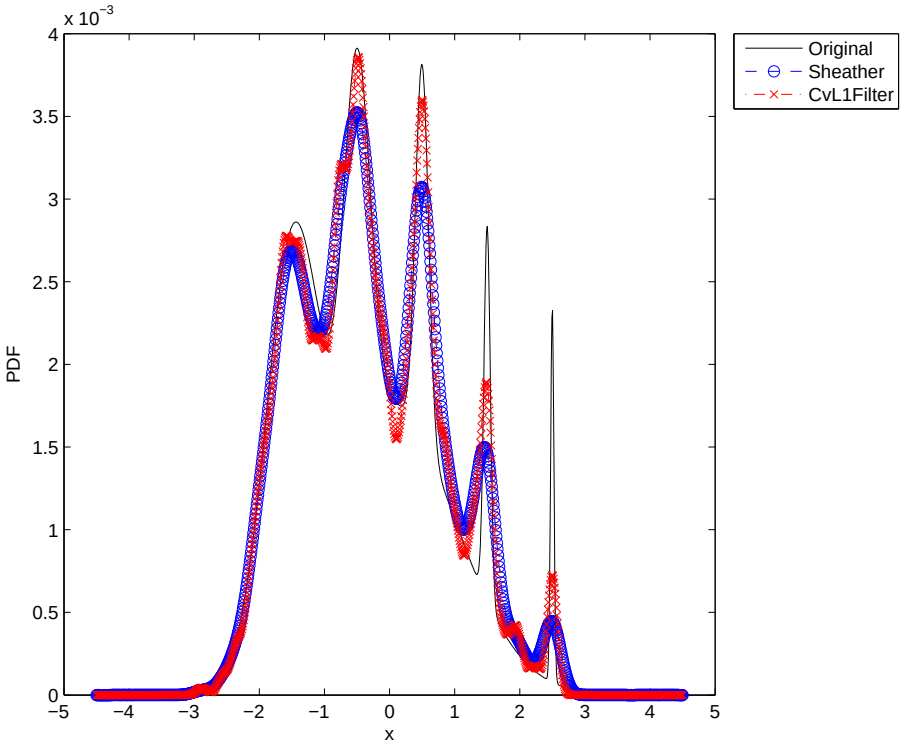

(i) Distribution 9



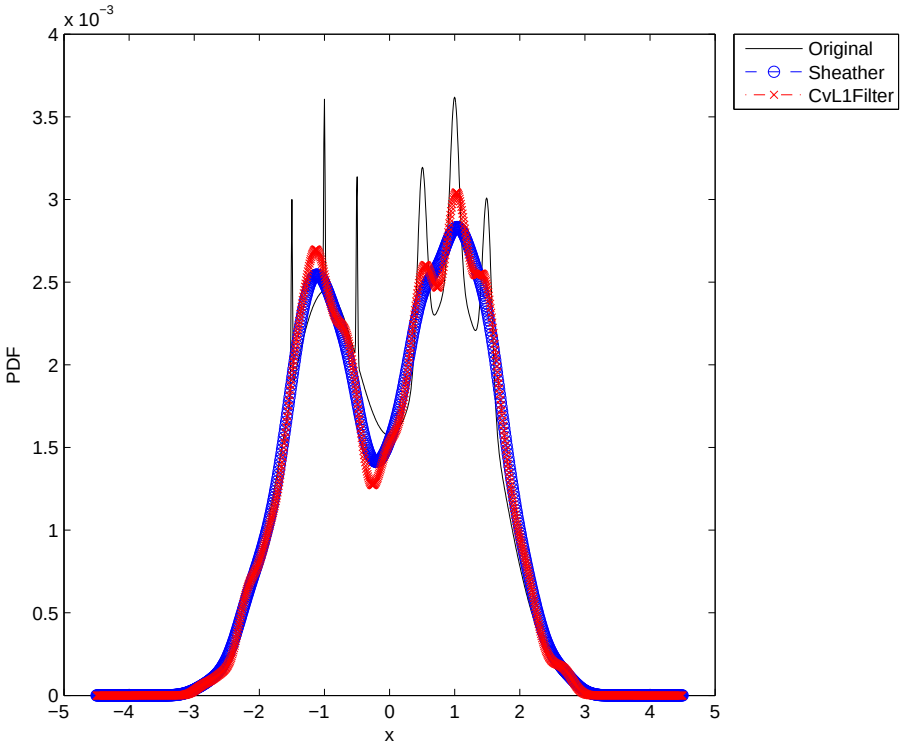

(j) Distribution 10



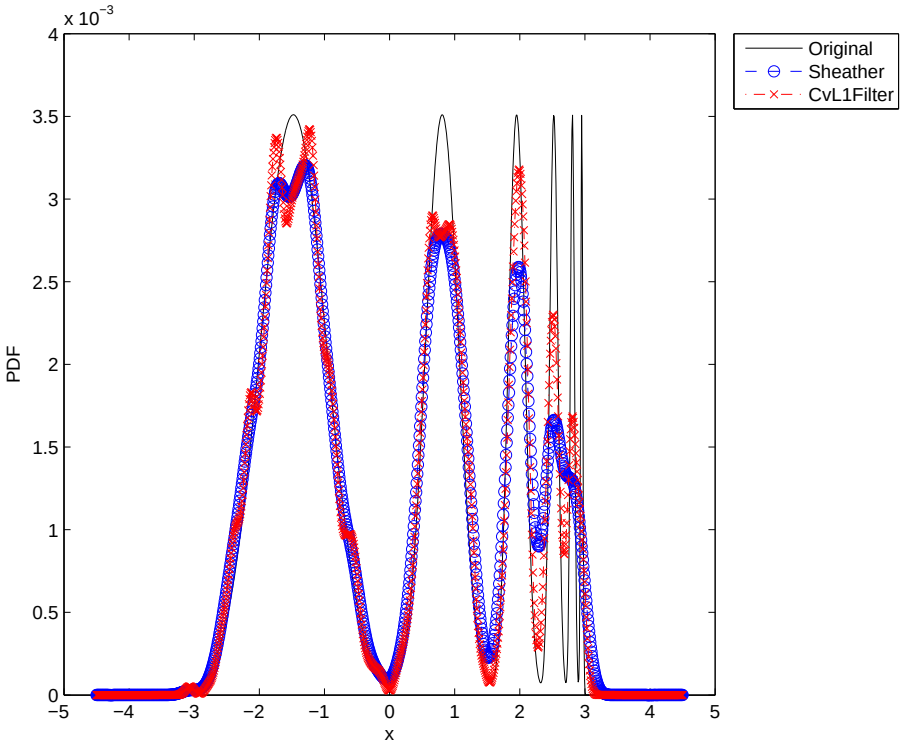

(k) Distribution 11



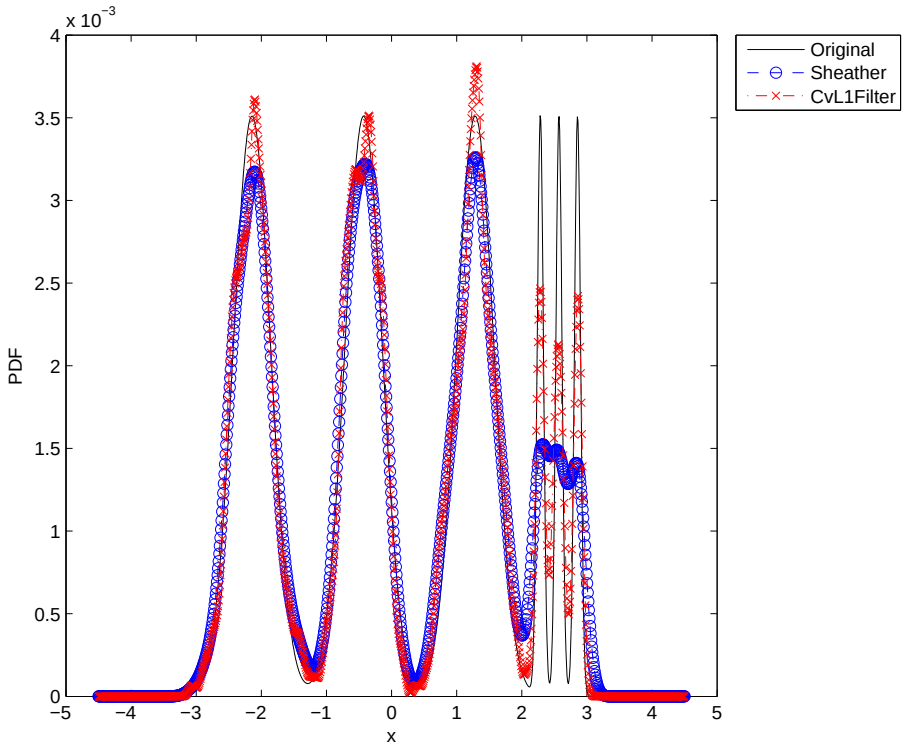

(l) Distribution 12




(m) Distribution 13




(n) Distribution 14




(o) Distribution 15

Figure 4.2: Results of KDE for 15 example distributions used in this chapter. Shown are the original (black), KDE with Sheather's method (blue) and KDE with the proposed method cross validation (red) and the proposed cross validation with $l_1$ norm term and filter (green).

# Chapter 5

# IMAGE THRESHOLDING USING MIXTURES OF GAUSSIANS

One of the most basic tasks in image processing is thresholding. In the simplest scenario the image can be divided into two clusters: foreground and background. To segment the image accordingly, one needs to find a threshold for the image intensities [77], [78]. This chapter focuses on global thresholding. Adaptive thresholding as in [79] or [80] will not be considered. Reference [34] provides an in-depth look at different techniques for the single-threshold scenario. It also provides several metrics with which one can measure the performance of a given method. A very popular approach is Otsu's thresholding [36], in which the intra-class variance is minimized by means of an iterative exhaustive search.

In this chapter, we introduce a different approach to thresholding. Our method relies on an iterative algorithm that estimates a PDF based on Kernel Density Estimation (KDE) [27] using a Gaussian kernel for the two clusters. The influence of the Gaussian mixtures is measured for the current value of the

image intensity histogram which is assigned to one of the two clusters according to the iteratively estimated likelihood values of the two clusters. This is continued until the current image intensity value is assigned to a different class than its predecessor.

The details of the algorithm are explained in Section 5.1. The algorithm is guaranteed to converge to a solution and does not use an exhaustive search. In our experiments the segmentations obtained by our algorithm on average scored higher than Otsu's thresholding method. Experimental results are presented and discussed in Section 5.2.

## 5.1 Description of the Algorithm

For our threshold choice we iteratively compute two mixtures of Gaussians (one for each cluster) using the histogram of the data as the input. We evaluate the values of the mixtures for the two image intensity values that are next to the previously computed mixture of Gaussians and not yet assigned to a cluster. The intensity values are assigned to the very cluster whose likelihood estimate is larger at the given bin.

We estimate a PDF for each cluster in an iterative manner. The update step of the iterative algorithm works as follows: Let $h[k]$ denote the normalised $K$-point histogram. We assign the i-th level of the histogram to one of the clusters as follows: Let $p_j(k|L)$ and $p_j(k|U)$ be the currently estimated PDFs of the lower cluster and upper cluster, respectively. The subscript $j$ indicates the $j$-th iteration step. The PDFs are estimated using a Gaussian kernel. For the running index $i$, we compare the likelihoods

$$|l_j(i|L)| \gtrless |l_j(i|U)|\,, \tag{5.1}$$

where $l_j(i|L) = p_j(i|L)|_{k=i}$ and $l_j(i|U) = p_j(i|U)|_{k=i}$, respectively. If the left side of Equation 5.1 is larger than the right side, then $h[i]$ is assigned to the lower cluster, else it is assigned to the upper cluster. Let us define

$$s_i = \sum_{l=1}^{i} h[l] \neq 1, \tag{5.2}$$

where

$$s_i = s_{i-1} + h[i]. \tag{5.3}$$

$$p_{j+1}(k|L) = \frac{s_{i-1}}{s_i} p_j(k|L) + \frac{h[i]}{s_i} g(k; i, \sigma_L), \tag{5.4}$$

where $g(k; \mu, \sigma)$ denotes a Gaussian function at point $k$ with mean $\mu$ and standard deviation $\sigma$ according to

$$g(k; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(k-\mu)^2}{2\sigma^2}}. \tag{5.5}$$

The standard deviation of the Gaussian kernel function in the lower cluster is denoted by $\sigma_L$, while the standard deviation of the Gaussian kernel function in the upper cluster is denoted by $\sigma_U$, respectively. If the right side of Equation 5.1 is larger than the left side, the upper cluster is updated in a similar manner.

The current bin is assigned to the cluster whose likelihood function is larger at the given bin, according to Eq. 5.1. The process continues if the cluster of the current bin is the same as the last bin and $i$ is incremented by one, if not, the procedure is aborted and the arithmetic mean of the numbers $i$ and $i-1$ is the threshold that is outputted.

In Figure 5.1, the final stage of the algorithm is shown for a toy example. The bins that have previously been assigned to the lower cluster L are signified by having a circle on top, while the bins that have previously been assigned to the upper cluster U have a diamond on top. The dashed line of the likelihood funtion of cluster U is now higher at the fourth bin than the solid line denoting the likelihood function of cluster L, therefore bin 4 gets assigned to cluster U. Bin 4

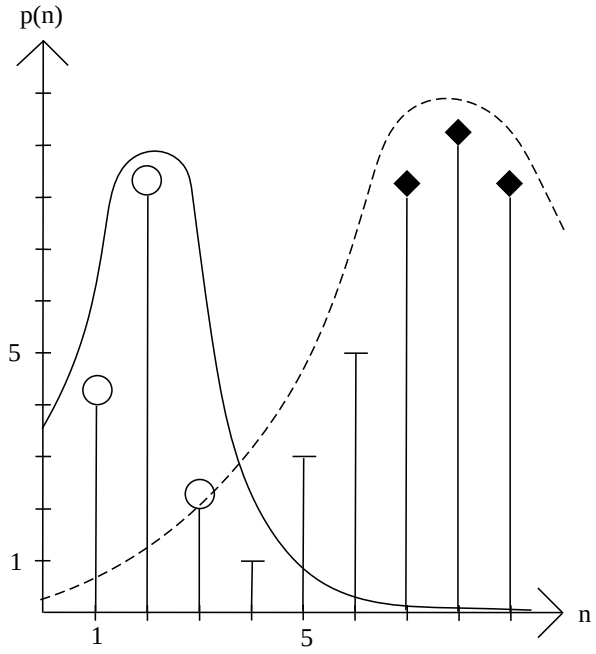



Figure 5.1: An illustration of the algorithm. The figure shows the final stage of the algorithm. Bin 4 is assigned to cluster U and the algorithm is aborted.

is assigned to a different cluster than its predecessor bin 3, thus, the algorithm is aborted and the value 3.5 is the estimated threshold.

### 5.1.1 Bandwidth choice of the Gaussian kernel

An important parameter of KDE is the estimation of kernel bandwidth or standard deviation $\sigma$. In this thresholding algorithm we can use a predefined bandwidth [27]. However, it is also possible to select a $\sigma$ in an adaptive manner. In this subsection, an adaptive variance estimation algorithm is described.

In the adaptive standard deviation estimation method, $\sigma$ values of the Gaussian kernel are computed for each cluster separately. In a given step of the algorithm, $\sigma_L$ is chosen according to

$$\begin{aligned} \min_{\sigma_L} \quad & \left| \frac{h[i]}{s_i} - p_{j+1}(i, L) \right| \\ \text{subject to} \quad & \sigma_{min} < \sigma_L < \sigma_{max}, \end{aligned} \tag{5.6}$$

where $\sigma_{min}$ and $\sigma_{max}$ are predefined thresholds. The estimate for $\sigma_U$ is chosen similarlily for the other cluster. Our experiments showed that the choice of the two free parameters $\sigma_{min}$ and $\sigma_{max}$ doesn't need any elaborate parameter tuning. Reasonable choices like $\sigma_{min} = 1$ and $\sigma_{max} = 25$ were used throughout our experiments and should produce decent results for other types of images as well. The estimated standard deviations of both clusters are used to construct the estimates of the cluster likelihoods according to Eq. 5.4.

It is also possible to use (4.11) for the estimation of $\sigma$. The results are shown in Table 5.2. When compared to the results using (5.6), both methods turn out to be comparable.

## 5.2 Experimental Results

Results are obtained over a dataset of 49 images and their segmentation groundtruth that was taken from [81] (19 images) and [82] (30 images). We quantify the performance of our method versus Otsu's method using the metrics proposed in [34] and [83], namely:

1. Region Non-Uniformity (RNU)
2. Misclassification Error (ME)
3. Relative Foreground Area Error (RFAE)
4. Edge Mismatch (EM)
5. Normalised Modified Hausdorff Distance (NMHD)
6. Normalised False Detection Ratio (NFDR)

Only the first measure doesn't require groundtruth information. All six metrics are normalised so that a score of 0 denotes a "perfect" segmentation, whereas

| Image # | RNU | ME | RFAE | EM | NMHD | NFDR |
|---|---|---|---|---|---|---|
| 1 | -0.16 | -3.72 | -1.12 | -0.19 | 0.03 | -0.15 |
| 2 | -1.52 | 0.48 | -39.82 | -0.43 | 16.14 | 12.16 |
| 3 | -0.13 | 13.48 | 31.67 | 0.04 | 1.03 | 5.52 |
| 4 | -0.36 | 0.02 | -5.55 | -0.07 | 8.20 | -7.06 |
| 5 | -0.02 | -0.07 | -0.37 | 0.04 | -0.02 | -0.27 |
| 6 | -1.75 | 5.23 | 40.02 | 0.08 | 6.07 | 11.60 |
| 7 | 0.06 | 10.00 | 11.74 | -0.09 | 0.30 | 2.55 |
| 8 | 3.80 | -9.21 | -35.04 | -0.19 | -0.23 | -5.43 |
| 9 | 1.44 | 10.07 | -9.36 | -0.87 | 25.44 | 21.20 |
| 10 | -0.06 | 0.10 | 0.55 | 0.02 | -0.01 | 0.16 |
| 11 | 0.14 | 1.14 | 0.62 | 0.02 | 0.01 | 0.16 |
| 12 | 0.38 | 21.30 | 10.56 | -0.18 | 0.87 | 0.75 |
| 13 | 0.42 | 34.14 | 21.07 | -0.13 | 0.32 | 2.98 |
| 14 | 0.02 | 61.87 | -13.59 | 0.98 | 22.00 | 1.25 |
| 15 | 0.24 | -2.31 | -10.30 | -0.03 | -0.54 | -5.53 |
| 16 | 4.32 | 23.96 | 11.86 | -0.21 | -0.35 | 3.07 |
| 17 | 0.64 | 5.16 | 7.58 | 0.07 | 0.09 | 2.60 |
| 18 | -1.51 | 13.50 | -26.83 | -0.11 | 4.28 | 46.54 |
| 19 | 0.28 | 4.40 | 1.71 | -0.02 | 0.15 | 0.34 |
| 20 | 4.23 | 55.10 | -5.47 | -0.82 | 1.63 | 63.49 |
| 21 | -0.42 | 22.91 | 37.10 | -0.08 | 0.76 | 29.14 |
| 22 | -0.97 | 9.48 | -9.91 | -0.05 | 3.29 | 48.76 |
| 23 | 0.99 | 29.24 | 7.89 | 0.08 | 1.00 | 67.66 |
| 24 | -1.30 | 2.74 | 37.42 | -0.42 | 0.16 | 12.36 |
| 25 | -0.04 | 0.52 | 0.82 | 0.00 | -0.03 | 0.19 |
| 26 | 1.03 | -1.77 | 3.21 | 1.02 | -0.39 | 2.85 |
| 27 | 0.05 | -0.04 | 0.29 | 4.75 | -0.02 | 0.24 |
| 28 | 4.06 | 5.76 | 39.46 | -0.98 | -2.16 | 40.39 |
| 29 | 0.20 | 0.49 | 1.44 | 1.05 | -0.10 | 0.94 |
| 30 | -0.85 | 1.88 | 24.77 | -0.23 | 5.19 | 27.42 |
| 31 | -1.44 | -3.68 | -58.44 | -24.20 | -33.42 | -8.21 |
| 32 | -0.51 | 1.43 | 18.61 | -0.08 | -0.45 | 4.42 |
| 33 | -0.08 | 7.64 | 10.29 | 0.07 | 0.26 | 2.45 |
| 34 | -0.15 | 5.61 | 11.03 | 0.00 | 0.37 | 2.73 |
| 35 | 1.43 | 16.58 | 28.40 | -0.22 | 0.16 | 17.50 |
| 36 | -0.68 | 3.56 | -18.97 | 0.00 | 0.96 | 21.36 |
| 37 | -0.46 | 14.32 | 38.68 | -0.08 | 2.63 | 9.71 |
| 38 | 2.22 | -12.18 | -10.98 | 0.04 | 0.55 | -1.22 |
| 39 | 0.66 | 8.25 | 15.05 | 0.00 | -0.12 | 4.80 |
| 40 | 2.20 | -5.71 | -4.77 | 0.06 | 0.15 | -0.41 |
| 41 | 0.75 | 10.56 | 10.84 | -0.12 | 0.26 | 1.42 |
| 42 | -0.43 | 6.03 | 4.85 | -0.01 | 0.01 | 0.27 |
| 43 | -0.06 | 1.56 | 3.12 | 0.02 | -0.09 | 1.03 |
| 44 | -0.43 | 32.55 | 34.55 | -0.10 | 1.13 | 4.31 |
| 45 | 0.02 | 2.68 | 0.50 | 0.00 | 0.02 | 0.02 |
| 46 | -0.26 | 0.08 | -17.38 | -0.30 | 7.51 | -49.66 |
| 47 | -0.67 | -0.59 | -23.84 | -0.59 | 27.41 | -40.78 |
| 48 | -0.81 | -3.35 | -15.71 | -1.91 | -6.59 | -0.40 |
| 49 | -1.01 | 2.35 | 8.91 | -0.06 | 0.39 | 0.84 |
| Mean | 0.28 | 8.24 | 3.41 | -0.50 | 1.92 | 7.27 |
| Success rate | 46.96 | 77.55 | 63.27 | 40.82 | 69.39 | 77.55 |

Table 5.1: Differences between Otsu's method and our proposed method for different distance metrics according to Equation (5.7). A positive value indicates that our method had a lower score and was therefore better than Otsu's method. The last row denotes the percentage of images in the dataset where our proposed methods outperforms Otsu's method.

a score of 1 denotes a completely incorrectly segmented image. For the modified Haussdorf distance, we used the implementation of [84] and the normalisation from [83]. Results can be seen in Table 6.2, where the differences of the metrics for Otsu's and our proposed algorithm are shown. Let $M_O$ denote the value of

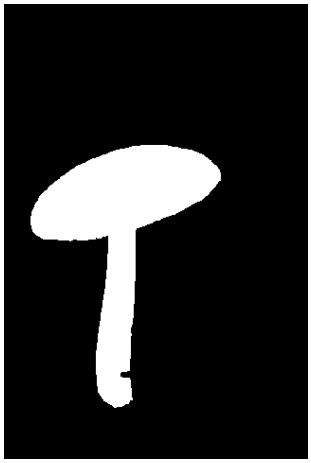
| Image # | RNU | ME | RFAE | EM | NMHD | NFDR |
|---|---|---|---|---|---|---|
| 1 | -0.13 | -2.85 | -0.86 | -0.05 | 0.03 | -0.11 |
| 2 | -1.60 | 0.19 | -41.94 | -0.47 | 16.87 | 7.94 |
| 3 | -0.12 | 15.31 | 47.20 | 0.00 | 1.91 | 8.61 |
| 4 | -0.75 | -0.31 | -10.68 | -0.12 | 4.12 | -12.23 |
| 5 | -0.02 | -0.07 | -0.37 | 0.04 | -0.02 | -0.27 |
| 6 | -1.91 | 5.35 | 40.31 | 0.08 | 6.85 | 14.47 |
| 7 | 0.04 | 16.25 | 24.84 | -0.15 | 0.61 | 5.53 |
| 8 | 2.85 | -5.76 | -26.26 | -0.20 | -0.24 | -4.21 |
| 9 | 0.91 | 9.86 | -39.63 | -1.53 | 12.74 | -10.18 |
| 10 | -0.11 | 0.19 | 0.98 | 0.01 | -0.01 | 0.29 |
| 11 | 0.20 | 1.54 | 0.86 | 0.03 | 0.01 | 0.22 |
| 12 | 0.24 | 28.31 | 18.77 | -0.21 | 1.54 | 1.35 |
| 13 | 0.07 | 45.98 | 48.00 | -0.14 | 0.63 | 7.21 |
| 14 | -0.97 | 61.67 | -15.66 | 1.17 | 20.30 | -1.17 |
| 15 | -0.25 | 1.87 | 2.69 | -0.01 | 0.60 | 6.59 |
| 16 | 4.58 | 27.84 | 14.81 | -0.27 | -0.32 | 3.71 |
| 17 | 0.70 | 5.48 | 8.05 | 0.07 | 0.11 | 2.75 |
| 18 | -2.10 | 13.51 | -42.22 | -0.13 | 5.20 | 16.99 |
| 19 | 0.41 | 6.83 | 2.75 | -0.02 | 0.23 | 0.55 |
| 20 | 4.04 | 57.32 | -15.52 | -0.83 | 2.20 | 36.95 |
| 21 | -0.54 | 25.88 | 16.87 | -0.09 | 1.19 | 48.51 |
| 22 | -1.25 | 10.32 | -24.08 | -0.05 | 4.30 | 73.33 |
| 23 | 0.94 | 33.69 | -6.26 | 0.04 | 0.99 | 43.28 |
| 24 | -1.78 | 2.64 | 13.61 | -0.48 | 0.08 | 26.00 |
| 25 | -0.13 | 1.72 | 2.81 | 0.01 | -0.10 | 0.67 |
| 26 | 1.94 | -0.63 | 6.05 | 1.53 | -0.55 | 4.70 |
| 27 | 0.05 | -0.04 | 0.29 | 4.75 | -0.02 | 0.24 |
| 28 | 4.10 | 5.23 | 47.14 | -0.91 | -2.46 | 48.32 |
| 29 | 0.25 | 0.65 | 1.64 | 1.24 | -0.11 | 1.06 |
| 30 | -0.90 | 1.55 | 19.33 | -0.25 | 5.62 | 33.98 |
| 31 | -2.33 | -5.66 | -89.88 | -24.56 | -72.56 | -8.21 |
| 32 | -0.87 | 2.44 | 43.75 | -0.10 | -0.81 | 9.68 |
| 33 | -0.13 | 12.08 | 20.23 | 0.07 | 0.37 | 4.97 |
| 34 | -0.64 | 14.00 | 45.83 | -0.03 | 1.94 | 15.10 |
| 35 | 1.57 | 19.58 | 35.53 | -0.25 | 0.20 | 21.69 |
| 36 | -1.00 | 4.89 | -27.13 | 0.00 | 1.54 | 35.03 |
| 37 | -0.73 | 16.98 | 56.53 | -0.11 | 4.50 | 18.55 |
| 38 | 1.98 | -10.65 | -10.02 | 0.05 | 0.49 | -1.12 |
| 39 | 0.83 | 12.65 | 29.25 | 0.00 | -0.26 | 9.07 |
| 40 | 1.82 | -4.70 | -4.09 | 0.09 | 0.13 | -0.36 |
| 41 | 1.26 | 25.29 | 46.37 | -0.16 | 1.43 | 6.00 |
| 42 | -0.54 | 7.79 | 6.76 | -0.02 | 0.02 | 0.38 |
| 43 | -0.06 | 2.22 | 4.56 | 0.02 | -0.12 | 1.50 |
| 44 | -0.77 | 41.66 | 70.66 | -0.22 | 2.03 | 12.52 |
| 45 | 0.08 | 26.10 | 8.76 | 0.04 | 0.20 | 0.30 |
| 46 | -0.56 | -4.83 | -45.98 | -0.49 | -2.31 | -67.73 |
| 47 | -0.67 | -1.18 | -26.25 | -0.81 | 24.30 | -40.93 |
| 48 | -0.87 | -4.14 | -19.41 | -2.17 | -9.00 | -0.45 |
| 49 | -1.09 | 2.51 | 10.00 | -0.06 | 0.40 | 0.98 |
| Mean | 0.12 | 10.75 | 5.08 | -0.52 | 0.71 | 7.80 |
| Success rate | 42.86 | 75.51 | 63.27 | 34.69 | 69.39 | 75.51 |

Table 5.2: Differences between Otsu's method and our proposed method for different distance metrics according to Equation (4.14). A positive value indicates that our method had a lower score and was therefore better than Otsu's method. The last row denotes the percentage of images in the dataset where our proposed methods outperforms Otsu's method.

one of the above mentioned metrics for Otsu's method and let $M_P$ denote the respective value for our proposed method. Then the residual

$$R = 100 \cdot (M_O - M_P) \tag{5.7}$$

(a) Original image



(b) Groundtruth

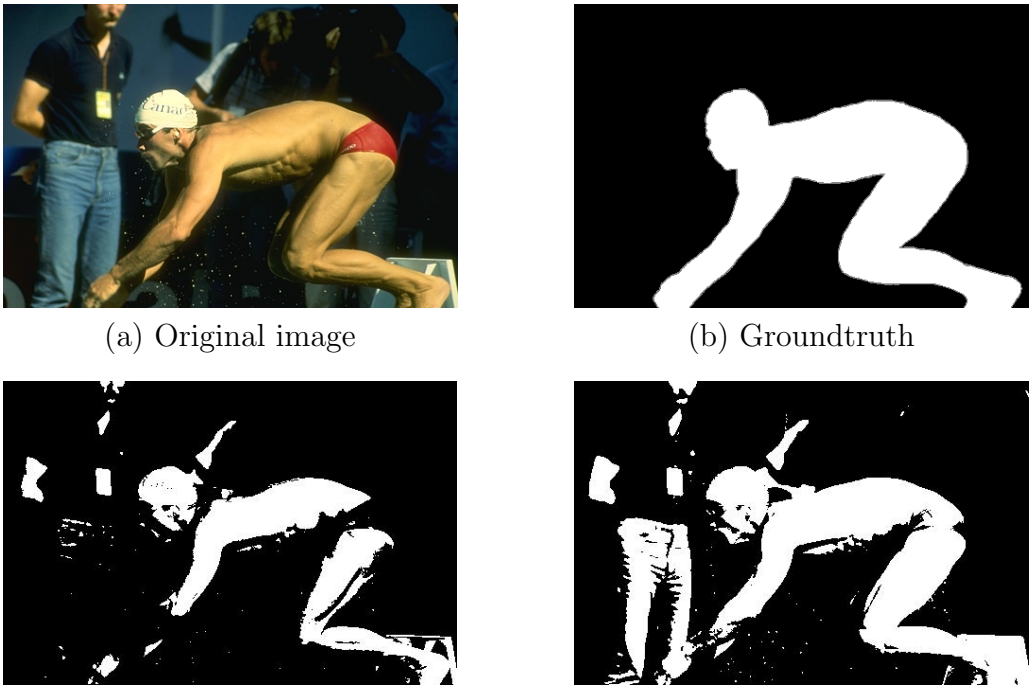
(c) Our segmentation

(d) Otsu's segmentation

Figure 5.2: Example in which our method performed better than Otsu's method according to five out of six metrics: (a) image 6 from the dataset, (b) its groundtruth segmentation, (c) our proposed segmentation and (d) Otsu's segmentation.

is the value shown in Table 6.2. A positive value denotes that our algorithm had a lower score than Otsu's and is therefore preferable. The mean values over the whole dataset are also given. Furthermore, the last row shows the percentage of images in the dataset where our method outperformed Otsu's method. On average, our method has 3% better performance than Otsu's method over the whole dataset. In 63% of the images from the dataset our method had a better score than Otsu's method. In Table 5.2 similar experiments were carried out, but here (4.11) from Chapter 4 was used for the $\sigma$ estimation. The upper and

lower bound of $\sigma$ was chosen as 5 and 2, respectively. On average, this method has 4% better performance than Otsu's method over the whole dataset. In 60% of the images from the dataset the $l_1$-based method had a better score than Otsu's method. There is no clear winner among the two proposed $\sigma$ estimation methods in thsi dataset. While the $l_1$ norm approach yields better performance on average, the adaptive approach works better for more images.



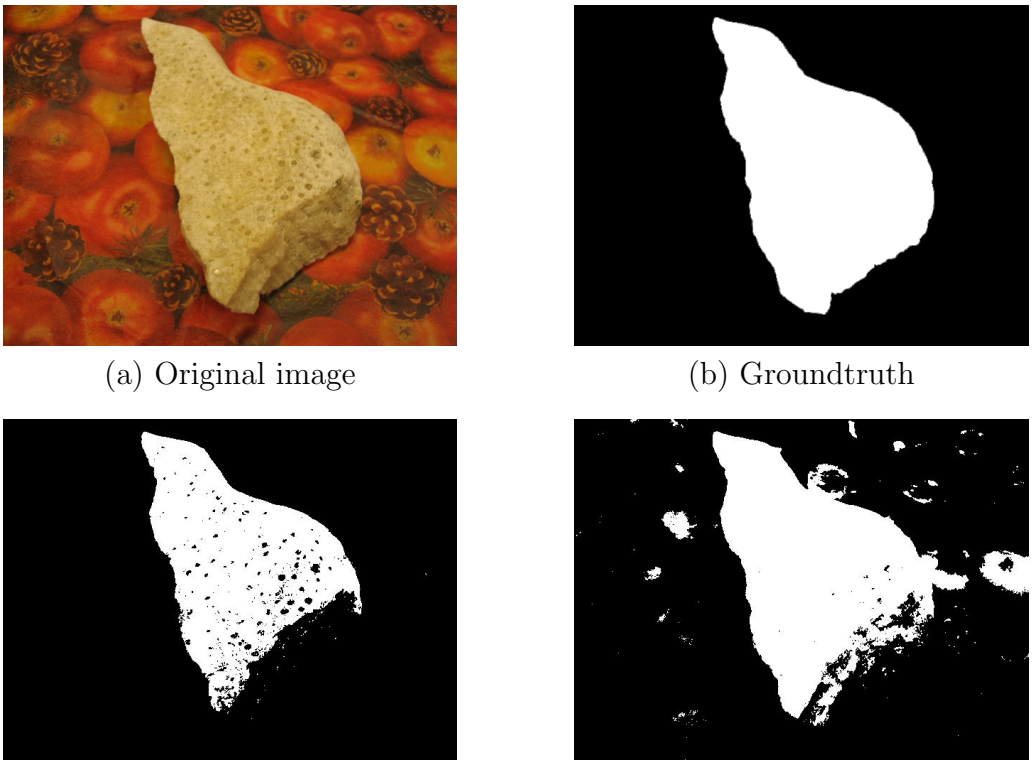
(a) Original image (b) Groundtruth

(c) Our segmentation (d) Otsu's segmentation

Figure 5.3: Example in which our method performed worse than Otsu's method according to three out of six metrics: (a) image 2 from the dataset, (b) its groundtruth segmentation, (c) our proposed segmentation and (d) Otsu's segmentation.

Out of the six tested metrics our algorithm on average outperformed Otsu's method in five cases, namely RNU, ME, RFAE, NMHD and NFDR. Furthermore, for ME, RFAE, NMHD and NFDR, our method outperformed Otsu's method in the majority of images in the dataset. The following discussion focuses on the $\sigma$ estimation from (5.6) given Section 5.1.1.

As can be seen in Figures 5.2 - 5.4, our method generally produced segmentations where the foreground is not as well connected compared to Otsu's method.



(a) Original image (b) Groundtruth

(c) Our segmentation (d) Otsu's segmentation

Figure 5.4: Example in which our method performed worse than Otsu's method according to five out of six metrics: (a) image 47 from the dataset, (b) its groundtruth segmentation, (c) our proposed segmentation and (d) Otsu's segmentation.

This was penalized by the region non-uniformity metric. An example can be seen in Figure 5.2. In this image, Otsu's method included more background regions into foreground compared to our method.

In Image 47 from our dataset, which is shown in Figure 5.4, respectively, Otsu's method is rated higher than our proposed method according to five out of the six studied metrics. However in case of Image 47, our method again picked up less background than Otsu's method and one can also see the texture of the object in our segmentation. In this case the foreground object had two levels. Our method rejected the darker levels but Otsu's method included them to the foreground at the expense of including some background object to the foreground. Similarily, in Image 2 shown in Figure 5.3, our method also picked

(a) Original image

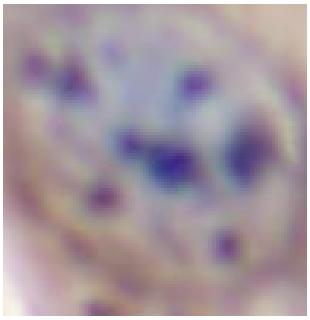
(b) Our segmentation

(c) Otsu's segmentation

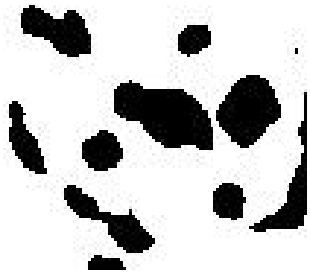
Figure 5.5: Example of a possible application area: (a) cropped cell image from the BT-20 cancer cell line, (b) our proposed segmentation and (c) Otsu's segmentation.

up less background than Otsu's method. Clearly, our method produced a better result by not including the pants of the referee in the foreground. In Image 2, our method was rated higher than Otsu's method in three out of six metics.

Figures 5.5-5.8 show a possible application area of our method. The figure show example images of nuclei from the cancer cell line BT-20. The number of black areas in the cytoplasm decreases as one moves away from the nuclei, therefore segmenting these areas in an efficent manner will be useful for the



(a) Original image

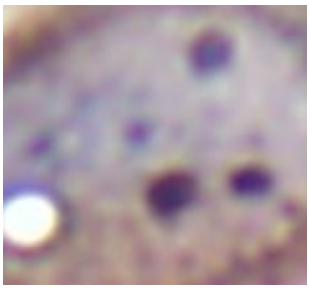


(b) Our segmentation

(c) Otsu's segmentation

Figure 5.6: Example of a possible application area: (a) cropped cell image from the BT-20 cancer cell line, (b) our proposed segmentation and (c) Otsu's segmentation.

analysis of microscopic images. As can be seen in Figures 5.5-5.8, our algorithm provides much better segmentations than Otsu's method.

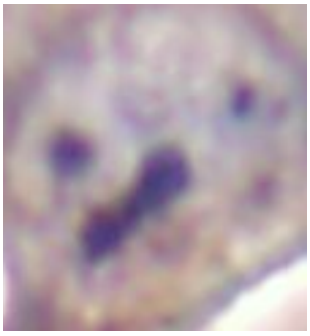


(a) Original image

(b) Our segmentation

(c) Otsu's segmentation

Figure 5.7: Example of a possible application area: (a) cropped cell image from a BT-20 cancer cell line, (b) our proposed segmentation and (c) Otsu's segmentation.



(a) Original image

(b) Our segmentation

(c) Otsu's segmentation

Figure 5.8: Example of a possible application area: (a) cropped cell image from the BT-20 cancer cell line, (b) our proposed segmentation and (c) Otsu's segmentation.

# Chapter 6

# CANCER CELL LINE CLASSIFICATION USING CO-DIFFERENCE OPERATORS

Implementation of signal processing algorithms usually requires a large number of multiplications. It is desirable to avoid heavy multiplication burden in applications where high energy consumption is a concern. Toward this end a new framework for signal processing is proposed by introducing a novel vector product definition that permits a multiplier-free implementation. The new vector product uses an operator that defines the product of two real numbers as the sum of their absolute values, with the same sign as the ordinary multiplication of two numbers. Multiplication is an energy consuming operation, especially in ARM-type processors used in tablet computers and mobile devices. Therefore, the new operator allows users to perform image and signal processing in low-cost processors.

An application of the new vector product is feature extraction from images. Recently, region covariance approaches [23] have become quite popular in the fields of image processing and computer vision. Their application includes object recognition, tracking and classification. Despite their superior performance, the computational burden of computing a covariance matrix for multiple image regions is relatively high. A computationally more efficient alternative to the covariance matrix, the so-called co-difference matrix was introduced [24] and successfully applied to several application areas [85]. In this chapter, firstly, we look more deeply into the properties of the co-difference operator. Secondly, we apply the co-difference matrix to a new application area, i.e., the classification of cancer cell line images and compare its performance with the established covariance matrix method.

In section 6.1, the proposed multiplier-less operator is defined. In section 6.2 the application of the proposed multiplier to an image classification problem is explained and comparisons of the proposed region co-difference matrix with the established region covariance matrix are shown.

## 6.1 Vector Product and Image Feature Extraction Algorithm

Given an intensity image I of size $m \times n$, we define a mapping $\phi$ from image domain to feature domain as

$$F(x, y) = \phi(I, x, y) \tag{6.1}$$

where each pixel (x,y) is mapped to a set of features and $F$ is a feature function computed as a $m \times n \times d$ array. For a given subwindow $W$ consisting of $N$ pixels, let $(\mathbf{f_k})_{k=1...n}$ be the $d$-dimensional feature vectors extracted from $W$. Then, the

covariance matrix of region $W$ can be computed as follows

$$\mathbf{C}_v = \frac{1}{N-1}\sum_{k=1}^{N}(\mathbf{f_k} - \boldsymbol{\mu}) \cdot (\mathbf{f_k} - \boldsymbol{\mu})^T \tag{6.2}$$

where $\boldsymbol{\mu}$ is the mean vector of the feature vectors inside the region $W$. The covariance matrix is symmetric positive-definite and of size $d$-by-$d$.The computational complexity of the covariance matrix is relatively large, i.e., $\mathcal{O}(N)$. Computational cost becomes especially important when a large image needs to be scanned at different scales. In [24], an efficient multiplier-less implementation of covariance descriptors, called co-difference matrices, was introduced. In the co-difference method, an operator based on additions instead of the multiplications as in the covariance method is used. This methodology gives comparable results in regular texture classification [85]. Let $a$ and $b$ be two real numbers. The new operator is defined as follows:

$$a \oplus b = sgn(c) \cdot (|a| + |b|), \tag{6.3}$$

where $sgn(c)$ is the sign of $c$ and

$$c = \begin{cases} a \cdot b & \text{if } a \cdot b \neq 0 \\ a + b & \text{if } a \cdot b = 0 \end{cases} \tag{6.4}$$

The operator is basically a summation operation but the sign of the results behaves like the multiplication operation. As a result, the computational complexity of signal analysis can be decreased significantly.

We measured the effect of this speed-up and show the results in Table 6.1. For each class of cancer cell lines we computed the covariance and co-difference matrices with the features that will be presented in Section 6.2.1. We measured the duration of the computations for both matrices as well as their ratio. Computations were carried out on a computer running a Windows 7 operating sytem with an AMD Phenom® II X6 1090T processor. We implemented both methods using MATLAB® with simple *for* loops. Absolute computation time can be increased with faster implementations, but the speed-up between the two

methods can thus be measured effectively. In summary, for a given image from our dataset, the computation of the region co-difference matrices for the full feature vector was about 100 times faster when compared to the region covariance matrices.

| Class | Covariance cost [s] | Co-difference cost [s] | Co-difference speed-up [/] |
|---|---|---|---|
| 1 | 646.63 | 6.02 | 107.43 |
| 2 | 761.17 | 7.16 | 106.25 |
| 3 | 746.21 | 7.08 | 105.33 |
| 4 | 380.06 | 3.57 | 106.60 |
| 5 | 928.97 | 8.85 | 104.96 |
| 6 | 1467.17 | 13.87 | 105.75 |
| 7 | 1444.33 | 13.70 | 105.42 |
| 8 | 501.49 | 4.66 | 107.51 |
| 9 | 535.87 | 5.04 | 106.25 |
| 10 | 474.77 | 4.43 | 107.22 |
| 11 | 536.97 | 5.15 | 104.17 |
| 12 | 1319.80 | 12.66 | 104.21 |
| 13 | 1174.06 | 11.19 | 104.88 |
| 14 | 592.35 | 5.61 | 105.66 |
| Mean | 822.13 | 7.79 | 105.83 |

Table 6.1: Covariance vs. co-difference computational cost. The fourth column shows the ratio of the second and third column, respectively.

We define a new "vector product" of two N-dimensional vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ in $R^N$ as follows:

$$< \mathbf{x}_1, \mathbf{x}_2 >= \sum_{i=1}^{N} x_1(i) \oplus x_2(i), \tag{6.5}$$

where $x_1(i)$ and $x_2(i)$ are the $i$-th entries of the vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively. The co-difference matrix can be constructed from the vector product of two vectors. One can also define a "vector product space" analogous to an inner product space though where our vector product satisfies the requirements of an inner product except the condition of linearity in the first argument. We define

the multiplication of a vector by a number $a$ based on (6.3) and (6.5) as follows:

$$a \oplus \mathbf{x} = \begin{bmatrix} a \oplus x(1) \\ a \oplus x(2) \\ \vdots \\ a \oplus x(N) \end{bmatrix}, \tag{6.6}$$

where $a$ is an arbitrary real number. Note that the vector product of a vector x with itself reduces to a scaled scaled $l_1$ norm of x as follows:

$$< \mathbf{x}, \mathbf{x} >= \sum_{i=1}^{N} x(i) \oplus x(i) = 2 \sum_{i=1}^{N} |x(i)| = 2||x||_1, \tag{6.7}$$

We define the co-difference matrix for the image region $R$ as

$$\mathbf{C}_d = \frac{1}{N-1} \sum_{k=1}^{N} (\mathbf{f_k} - \boldsymbol{\mu}) \oplus (\mathbf{f_k} - \boldsymbol{\mu})^T \tag{6.8}$$

which can replace the covariance matrix defined in Eq. (6.2) in image classification problems.

In practice, $f_k$ values can be color pixel values, wavelet transform coefficients, directional derivatives etc. In the next section, the codifference matrix is used in cancer cell-line image classification. A new directional correlation parameter constructed using the new vector product is also used as a feature parameter in covariance 6.2) and codifference matrices (6.8).

## 6.2 Experimental results

The new multiplication-free vector product is used in place of the conventional inner product and the efficiency of the proposed approach is demonstrated by a microscopic image classification example. Automatic classification of biomedical images is an emerging field, despite the fact that there is a long history of image recognition techniques [86]. Automated classification of carcinoma cells through morphological analysis will greatly improve and speed up cancer research

conducted using established cancer cell lines as in vitro models. Distinct morphologies of different types and even sub-types of cancer cells reflect, at least in part, the underlying biochemical differences, i.e., gene expression profiles. Moreover, the morphology of cancer cells can infer invasivenes of tumor cell and hence the metastatic capability.

In molecular biology studies, experimenters deal with a large number of specimens whose identity have to be checked recurringly during different stages of the experiment. Therefore, predicting labels of cancer cell lines in a fast and accurate manner via a pattern classification approach will greatly enhance biologists' ability to identify different types of cell lines without the need to scrutinize each and every microscopic image one by one. Although cell lines are being used widely as in vitro models in cancer research and drug development, mislabeling cell lines or failure to recognize any contamination may lead to misleading results. Short tandem repeat (STR) analysis is being used as a standard for the authentication of human cell lines. However, this process takes a long time and has to be carried out by an expert. Automated analysis, on the other hand, will provide the scientists a fast and easy-to-use tool that they can use in their own laboratories to verify their cell lines.

Modelling of cell morphology has been studied by several groups, for example for fission yeast in [87] and for e. coli bacteria in [88]. In the fission yeast case, differential expression of protein affects the cell size and, therefore, cell fate, while in the e. coli case, the topological organization is analyzed with respect to the underlying signaling network. To the best of our knowledge there have been no studies that have used morphology of different human cancer cell lines for classification. Eventually, discrimination of 14 classes of biomedical images is achieved, which are all images of cancer cell lines. The dataset at hand consists of two major types of cancer cell lines, namely breast cancer and liver cancer

(hepatocellular carcinoma) with 7 sub-classes, respectively. The dataset consists of 280 images, i.e., 20 per sub-class.

Our approach aims to carry out the automated analysis by extracting a feature vector from the images. These feature parameters reflect the large morphological diversity of the images. The six hepatocellular carcinoma, one hepatoblastoma and seven breast cancer cell lines were obtained from the following sources: FOCUS ( [89]), Hep40 ( [90]), Huh7 (JCRB JCRB0403), Mahlavu ( [91]), PLC (ATCC CRL-8024), SkHep1 (ATCC HTB-52), HepG2 (ATCC HB-8065), BT-20 (ATCC HTB-19), CAMA-1 (ATCC HTB-21), MDA-MB-157 (ATCC HTB-24), MDA-MB-361 (ATCC HTB-27), MDA-MB-453 (ATCC HTB-131), MDA-MB-468 (ATCC HTB-132), T47D (ATCC HTB-133).

The cell lines were seeded into dishes with 20% confluency and grown at $37^oC$ under 5% $CO_2$ in standard Dulbecco's modified Eagle's medium (DMEM) supplemented with 10% FBS, 1% Non-Essential Aminoacid and 1% penicillin/streptomycin (GIBCO Invitrogen) up to 70% confluency. The authentication of the cell lines was regularly checked by STR profiling. Cell line images were obtained using an Olympus CKX41 inverted microscope with an Olympus DP72 camera with a 20x objective and recorded using 20x magnification. Some example images of different cancer cell line images can be seen in Figure 6.1.

### 6.2.1 Feature Extraction

Microscopic cancer cell line images contain significant amount of oriented singularities. As a result we used directional feature parameters constructed using the new vector product and the directional dual tree complex wavelet transform (DT-$\mathbb{C}$WT) for image representation.

### Directional Differences

In order to account for the large morphological variation of the images in our dataset, we evaluated differences between pixels in various directions. Consider two pixels $I(x, y)$ and $I(x + d \cdot \cos\alpha, y + d \cdot \sin\alpha)$ in an image $I$. Clearly, $I(x + d \cdot \cos\alpha, y + d \cdot \sin\alpha)$ lies on a circle, whose center point is $I(x, y)$ and has a radius $d$. Let us construct a vector of size $i = 1, 2, ..., A$ as follows:

$$\mathbf{I}_\alpha = [I_\alpha(1), I_\alpha(2), ..., I_\alpha(A)], \tag{6.9}$$

where $I_\alpha(i) = I(x + i \cdot \frac{R}{A} cos(\alpha), y + i \cdot \frac{R}{A} sin(\alpha))$, $\alpha$ is a specified angle and $R$ is a chosen radius. Then we can perform the introduced vector product according to (6.6) as follows:

$$\mathbf{s}_\alpha = I(x, y) \oplus (\mathbf{I}_\alpha - \mu_\alpha), \tag{6.10}$$

where $\mu_\alpha$ is the scalar mean value of $\mathbf{I}_\alpha$. $\mathbf{s}_\alpha$ values can be used as a directional feature set for classification purposes.

### Dual-Tree Complex Wavelet Transform

The dual-tree complex wavelet transform (DT-$\mathbb{C}$WT) has been recently used in various signal and image processing applications [92], [93], [94] and [95]. It has desirable properties such as shift invariance, directional selectivity and lack of aliasing, other than conventional wavelet transforms [96], [97]. In the dual-tree $\mathbb{C}$WT, two maximally decimated discrete wavelet transforms are executed in parallel, where the wavelet functions of two different trees form an approximate Hilbert transform pair [98]. Two-dimensional DT-$\mathbb{C}$WT is also directionally selective in six different orientations. We use DT-$\mathbb{C}$WT complex coefficient magnitudes in detail subbands as pixel features and compute codifference descriptors.

Let $W_\theta^R(x, y)$ and $W_\theta^{Im}(x, y)$ denote, respectively, the real and imaginary part of the $2^{\text{nd}}$ level complex wavelet coefficient at the position (x,y) corresponding

to directional detail subbands at orientation $\theta$, where $\theta \in \{\pm 15^o, \pm 45^o, \pm 75^o\}$. The magnitude of the complex wavelet coefficent is then $M_\theta$, computed for $\theta \in \{\pm 15^o, \pm 45^o, \pm 75^o\}$. Hence, for each pixel in the average image $I_a(x, y)$, six complex wavelet coefficient magnitudes $M_\theta(x, y)$ representing six different orientations of DT-$\mathbb{C}$WT are extracted. These magnitudes will be utilized as features in the co-difference and covariance matrix computation for randomly sampled regions of the image $I_a(x, y)$.

**Feature vector choice**

With

$$\mathbf{M}_\theta(\mathbf{x}, \mathbf{y}) = [M_{\theta_1}(x, y)...M_{\theta_6}(x, y)] \tag{6.11}$$

and

$$\mathbf{S}_\alpha(\mathbf{x}, \mathbf{y}) = [\mathbf{s}_{\alpha_1}(x, y) \ ... \ \mathbf{s}_{\alpha_8}(x, y)] \tag{6.12}$$

where $\theta_1...\theta_6$ correspond to the six orientations of DT-$\mathbb{C}$WT detail subbands $\{\pm 15^o, \pm 45^o, \pm 75^o\}$.Parameters $\alpha_1...\alpha_8$ correspond to the mean of the eight angles of directional difference score estimation. The feature mapping function employed in this study is then

$$\phi(I, x, y) = [I_a(x, y) \, |I_x| \, |I_y| \, |I_{xx}| \, |I_{yy}| \, \mathbf{M}_\theta(\mathbf{x}, \mathbf{y}) \, \mathbf{S}_\alpha(\mathbf{x}, \mathbf{y})]^T, \tag{6.13}$$

where $|I_x|$ and $|I_{xx}|$ denote the first- and second-order derivatives at $(x, y)$ of the image $I_a$.

In this study, the performance of covariance and co-difference matrices is compared to their normalised counterparts given by

$$\hat{\mathbf{C}}(\mathbf{i}, \mathbf{j}) = \begin{cases} \sqrt{\mathbf{C(i, j)}}, & \text{if } i = j \\ \frac{\mathbf{C(i,j)}}{\sqrt{\mathbf{C(i,i)C(j,j)}}}, & \text{otherwise} \end{cases} \tag{6.14}$$

as in [99], [100] and [101].

### Foreground-Background Segmentation

The images in our dataset show a large amount of background pixels. Clearly, the background is not discriminative. Therefore, we address the issue of segmenting the images into foreground and background before classification. For our dataset, a simple thresholding scheme is not sufficient for segmentation, since foreground pixels have a large variance and may therefore have values higher and lower than the background pixels. We modeled the image as a mixture of two Gaussians, representing the foreground and background pixels, respectively. Using this model, an Expectation-Maximization (EM) algorithm was applied for segmentation. The result is noisy, so a morphological closing operation was applied, followed by median filtering. We obtained the sizes of the closing and median filter kernels by comparing the scores of the segmentation results of various kernel sizes. The used score was first described in [102] and evaluated in [103].

Since it is necessary to focus on foreground-like regions in carcinoma cell line images, $s$ analysis square windows are randomly selected, as in [104], from each image with the two constraints: the percentage of the foreground pixels in the selected region of an image must be above 50 and the variance of the selected region must exceed an image-dependent threshold, which is the variance of the whole image.

### Classification

For each subwindow, a codifference matrix was computed using (6.2) and the feature vector from (6.13). The image signature is composed of $s$ co-difference matrices of the same size. Each class is represented by $s\times\#$(images in each class) covariance matrices. A multiclass support vector machine (SVM) classifier is trained with an RBF kernel in the $d(d+1)/2$-dimensional vector space using the training points. SVM algorithm is implemented using LIBSVM library [71].

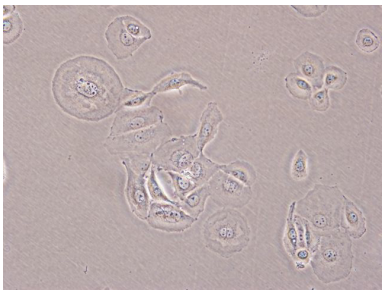
For each test subwindow, the corresonding co-difference descriptor is vectorized and fed into the trained SVM model for prediction. Therefore, there exist $s$ labels for each microscopic image corresponding to $s$ subwindows, and the image in question is assigned the label that gets the majority of votes among $s$ labels. The above process is carried out for unnormalised and normalised co-difference matrices.

**Results**

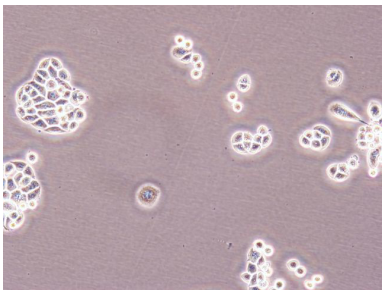
In our experiments, we compared the co-difference matrix with the covariance matrix of our features. When both were fed into an SVM, we got the results listed in Table 6.2. Clearly, the results are comparable, with the co-difference matrix outperforming the covariance matrix in the normalised case. The parameters for the SVM were $C = 1000$ and $\gamma = 0.5$ for covariance and co-difference matrices.

| Unnormalised Covariance | Normalised Covariance | Unnormalised Co-difference | Normalised Co-difference |
|---|---|---|---|
| 97.1 | 97.5 | 96.4 | 97.9 |

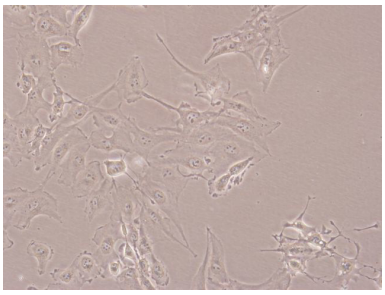
Table 6.2: Classification accuracies (in %) for different classifier inputs.



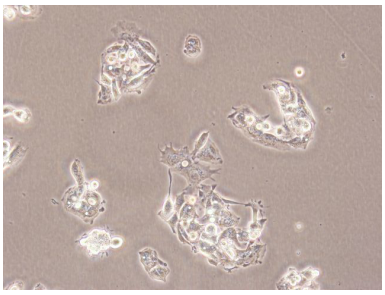
(a) BT-20 class



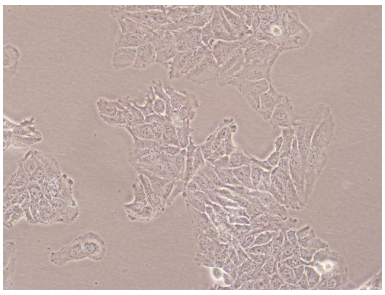
(b) Cama-1 class



(c) Focus class



(d) HepG2 class



(e) huh7 class

Figure 6.1: Examples of the image classes used in our experiments. Figure reproduced from Reference [4] with permission.

# Chapter 7

# CONCLUSIONS

In this thesis, we have studied several microscopic image processing modalities and resulting problems. We have proposed several methods adressing open problems in the image processing pipeline that follows image acquisition. Among those problems were image compression, analysis and classification.

In Chapter 2, we have developed an efficient compression algorithm that can be incorporated into the popular JPEG scheme. Our motivation here was the specific color content of stained microscopic images. For this reason we introduced an image-specific color transform based on the color content of a given image. The transform coefficients were adaptively computed for each image block. The first row of the transform matrix was determined by the color component ratios of previously compressed image blocks. Our experiments suggest that when this transform is used in standard JPEG, it results in higher PSNR for a given CR than standard colorspace transforms in general. Due to its conceptual simplicity and computational efficiency, our method can also be used in video compression. An application where the suggested method will be especially useful is microscopic images, in which the color palette is limited due to the staining process.

Our contributions to computer-aided diagnosis are in the classification of different follicular lymphoma grades and cancer cell lines on an image level. In Chapter 3, we demonstrated that the region covariance matrix can be used for automatic classification of follicular lymphoma images in a two-stage classification framework, in which the first stage is evaluating simple histograms from simple features and the second stage takes advantage of the sparsity of likelihood functions that are used in a Bayesian set-up. The classification method used in this chapter yields better results than the straightforward application of an SVM classifier.

In order to design more efficient classifiers, we looked more deeply into the topic of sparsity constraints for probability density function estimates. Those estimates can be used in Bayesian classification frameworks. In Chapter 4, we developed alternative ways to estimate the bandwidth in kernel density estimation. The proposed methods were compared with the commonly method by Sheather *et al.* [31]. On average, the total variation and filtered variation methods were found to result in higher fidelity than Sheather's method when measured under mean integrated square error and Kullback-Leibler divergence. The proposed methods are especially effective when the distribution to estimate is strongly multimodal. Since they utilize fast Fourier transform and fast converging convex set iterations, the proposed methods are of low algorithmic complexity.

We applied our proposed classification methods to the problem of image segmentation that frequently arises in the area of microscopic images, for example when cell nuclei need to be differentiated from cytoplasm. In Chapter 5, a novel algorithm for threshold determination in images was introduced. In the proposed method we estimated likelihood functions from the histogram of the data with a Gaussian kernel. Estimates of the free $\sigma$ parameter of the Gaussian function were obtained by carrying out a simple minimization over the data. The iterative algorithm is guaranteed to converge and does not require an exhaustive search.

We evaluated the efficiency of our algorithm using six metrics, out of which five vote our method as superior to Otsu's method.

In Chapter 6, a new framework for signal processing based on a novel vector product definition that permits a multiplier-free implementation was introduced. The main advantage of this framework is that it yields multiplication-free computationally efficient algorithms for performing some important tasks in image processing. This operator can be used to construct a so-called region co-difference matrix that has very similar properties to the established region covariance matrix. The co-difference matrix was successfully applied to the problem of classifiying cancer cell line images. The co-difference matrix-based approach produces slightly better results than the covariance matrix while being approximately 100-times faster. Its computational efficiency is of special significance for the practitioners in medicine and molecular biology, since faster processing times result in higher turnover of research results.

# Bibliography

[1] A. Suhre, K. Kose, M. N. Gurcan, and A. E. Cetin, "Content-adaptive color transform for image compression," *Optical Engineering*, vol. 50, no. 5, pp. 057003–057003–7, 2011.

[2] O. Lezoray and H. Cardot, "Cooperation of color pixel classification schemes and color watershed: a study for microscopic images," *IEEE Transactions on Image Processing*, vol. 11, pp. 783–789, July 2002.

[3] A. Suhre, T. Ersahin, R. Cetin-Atalay, and A. E. Cetin, "Microscopic image classification using sparsity in a transfer domain and Bayesian learning," in *Proceedings of 19th European Signal Processing Conference (EUSIPCO '11)*, pp. 1005–1009, 2011.

[4] F. Keskin, A. Suhre, K. Kose, T. Ersahin, A. E. Cetin, and R. Cetin-Atalay, "Image classification of human carcinoma cells using complex wavelet-based covariance descriptors," *PLoS ONE*, vol. 8, no. 1, p. e52807, 2013.

[5] R. Hooke, *Micrographia: or, Some physiological descriptions of minute bodies made by magnifying glasses.* J. Martyn and J. Allestry, London, 1665.

[6] E. Abbe, "Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung," *Archiv für mikroskopische Anatomie*, vol. 9, pp. 413–418, 1873.

[7] T. A. Klar, S. Jakobs, M. Dyba, A. Egner, and S. W. Hell, "Fluorescence microscopy with diffraction resolution barrier broken by stimulated emission," *Proceedings of the National Academy of Sciences*, vol. 97, no. 15, pp. 8206–8210, 2000.

[8] O. Heimstaedt, "Das Fluoreszenzmikroskop," *Z. Wiss. Mikrosk.*, vol. 28, pp. 330–337, 1911.

[9] F. Zernike, "Phase contrast, a new method for the microscopic observation of transparent objects part ii," *Physica*, vol. 9, pp. 974–986, dec 1942.

[10] M. Minsky, "Microscopy apparatus," *US Patent 3013467*, 1961.

[11] V. Prasad, D. Semwogerere, and E. Weeks, "Confocal microscopy of colloids," *Journal of Physics: Condensed Matter*, vol. 19, no. 11, pp. 113–102, 2007.

[12] D. Agard and J. Sedat, "Three-dimensional architecture of a polytene nucleus," 1983.

[13] D. Evanko, "Nature milestones in light microscopy," 2009.

[14] M. Hilbert and P. Lopez, "The worlds technological capacity to store, communicate, and compute information," *Science*, vol. 332, no. 6025, pp. 60–65, 2011.

[15] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, pp. 30–44, Apr. 1991.

[16] D. Penney, "A brief history of the biological stain commission: its founders, its mission and the first 75 years.," 2007.

[17] A. Meacutendez, P. Tahoces, M. Lado, M. Souto, and J. Vidal, "Computer-aided diagnosis: automatic detection of malignant masses in digitized mammograms," *Med Phys*, vol. 25, no. 6, pp. 957–64, 1998.

[18] M. N. Gurcan, Y. Yardimci, A. E. Cetin, and R. Ansari, "Automated detection and enhancement of microcalcifications in mammograms using nonlinear subband decomposition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97),*, vol. 4, pp. 3069 –3072 vol.4, apr 1997.

[19] J. Tang, R. Rangayyan, J. Xu, I. El Naqa, and Y. Yang, "Computer-aided detection and diagnosis of breast cancer with mammography: recent advances," *Trans. Info. Tech. Biomed.*, vol. 13, no. 2, pp. 236–251, 2009.

[20] M. Gurcan, L. Boucheron, A. Can, A. Madabhushi, N. Rajpoot, and B. Yener, "Histopathological image analysis: A review," *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 147 –171, 2009.

[21] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[22] K. P. Burnham, D. R. Anderson, and K. P. Burnham, *Model selection and multimodel inference: A practical information-theoretic approach*. Springer, 2nd ed., July 2002.

[23] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proceedings of European Conference on Computer Vision, Vol. 2*, pp. 589–600, 2006.

[24] H. Tuna, I. Onaran, and A. E. Cetin, "Image description using a multiplier-less operator," *IEEE Signal Processing Letters*, vol. 16, pp. 751 –753, September 2009.

[25] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2. ed., 2001.

[26] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[27] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, vol. 37. Chapman and Hall, 1986.

[28] O. Sertel, G. Lozanski, A. Shana'ah, and M. Gurcan, "Computer-aided detection of centroblasts for follicular lymphoma grading using adaptive likelihood-based cell segmentation.," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 10, pp. 2613–6, 2010.

[29] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[30] R. G. Baraniuk, "Compressive sensing," *Lecture Notes in IEEE Signal Processing Magazine*, vol. 24, pp. 118–120, Jul. 2007.

[31] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 53, no. 3, pp. pp. 683–690, 1991.

[32] M. C. Jones, J. S. Marron, and S. J. Sheather, "A brief survey of bandwidth selection for density estimation," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. pp. 401–407, 1996.

[33] C. R. Loader, "Bandwidth selection: classical or plug-in?," *The Annals of Statistics*, vol. 27, no. 27, pp. 415–438, 1999.

[34] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electron. Imaging.*, vol. 13, pp. 146 –168, jan. 2004.

[35] G. P. Balasubramanian, E. Saber, V. Misic, E. Peskin, and M. Shaw, "Unsupervised color image segmentation using a dynamic color gradient thresholding algorithm," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6806 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Mar. 2008.

[36] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62 –66, Jan. 1979.

[37] E. Gershikov, E. Lavi-Burlak, and M. Porat, "Correlation-based approach to color image compression," *Image Commun.*, vol. 22, pp. 719–733, Oct. 2007.

[38] Y. Chen, P. Hao, and A. Dang, "Optimal transform in perceptually uniform color space and its application in image coding," in *ICIAR (1)*, pp. 269–276, 2004.

[39] N.-X. Lian, V. Zagorodnov, and Y.-P. Tan, "Edge-preserving image denoising via optimal color space projection," *IEEE Transactions on Image Processing*, vol. 15, pp. 2575–2587, 2006.

[40] A. Okumura, J. Suzuki, S. Furukawa, I. Onu, and T. Ashihara, "Signal analysis and compression performance evaluation of pathelogical microscopic images," *IEEE Transactions on Medical Imaging*, vol. 16, pp. 701–710, 1997.

[41] T. F. T. Nakachi and T. Fujii, "Pathological microscopic image compression using lossless and progressive coding schemes," *Proc. of SPIE*, vol. 4323, pp. 413–122, 2001.

[42] A. Manduca, "Compressing images with wavelet/subband coding," *Engineering in Medicine and Biology, IEEE*, vol. 14, pp. 639–646, 1995.

[43] A. Manduca, "Medical image compression with set partitioning in hierarchical trees," *Proceedings of the 18th Annual International Conference of the IEEE*, vol. 3, pp. 1224–1225, 1996.

[44] A. M. V. Arya and R. Joshi, "An efficient coding method for teleconferencing and medical image sequences,," *Proc. on Third International Conference on Intelligent Sensing and Information Processing, 2005. ICISIP 2005*, vol. 93, pp. 8–13, 2005.

[45] S. H. Oguz, O. N. Gerek, and A. E. Cetin, "Motion-compensated prediction based algorithm for medical image sequence compression," *Signal Processing: Image Communication*, vol. 7, no. 3, pp. 225 – 230, 1995.

[46] M. J. Gormish, E. L. Schwartz, A. F. Keith, M. P. Boliek, and A. Zandi, "Lossless and nearly lossless compression for high-quality images," in *Proc. SPIE Vol. 3025, p. 62-70, Very High Resolution and Quality Imaging II, V. Ralph Algazi; Sadayasu Ono; Andrew G. Tescher; Eds.* (V. R. Algazi, S. Ono, and A. G. Tescher, eds.), vol. 3025 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pp. 62–70, Mar. 1997.

[47] G. Zamora, S. Yang, S. Mitra, M. Peterson, and E. P. Paschalis, "Efficient visualization of encoded fourier transform infrared microscopic data of osteoporotic bone," in *CBMS '01: Proceedings of the Fourteenth IEEE Symposium on Computer-Based Medical Systems*, (Washington, DC, USA), p. 197, IEEE Computer Society, 2001.

[48] H. I. Cuce, A. E. Cetin, and M. K. Davey, "Compression of images inCFA format," in *Image Processing, 2006 IEEE International Conference on*, pp. 1141 –1144, oct. 2006.

[49] O. N. Gerek and A. E. Cetin, "Adaptive polyphase subband decomposition structures for image compression," *IEEE Transactions on Image Processing*, vol. 9, pp. 1649–1660, 2000.

[50] O. N. Gerek and A. E. Cetin, "A 2-d orientation-adaptive prediction filter in lifting structures for image coding," *IEEE Transactions on Image Processing*, vol. 15, pp. 106–111, Jan. 2006.

[51] H. Trussell and M. Vrhel, *Foundations of Digital Imaging.* Cambridge Press, 2008.

[52] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE Transactions on Image Processing*, vol. 6, pp. 901–932, July 1997.

[53] E. Hamilton, "Jpeg file interchange format (version 1.02)," Sept. 1992.

[54] N. Jayant and P. Noll, *Digital Coding of Waveforms - Principles and Applications to Speech and Video*, p. 688. Taipei, Taiwan: Kai Fa Book Company, 1990.

[55] Kodak Corporation, "Kodak dataset," Nov. 1999.

[56] E. Gershikov and M. Porat, "On color transforms and bit allocation for optimal subband image compression," *Image Commun.*, vol. 22, pp. 1–18, Jan. 2007.

[57] Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Computer Graphics and Image Processing*, vol. 13, pp. 222 – 241, July 1980.

[58] W. Pratt, "Spatial transform coding of color images," *IEEE. Trans. Commun.*, vol. 19, pp. 980–992, 1971.

[59] T. Moon and W. Stirling, *Mathematical Methods and Algorithms for Signal Processing.* Prentice-Hall, 2000.

[60] D. Marpe, H. Kirchhoffer, V. George, P. Kauff, and T. Wiegand, "An adaptive color transform approach and its application in 4:4:4 video coding," in *Proc. 14th European Signal Processing Conference (EUSIPCO 06), Florence, Italy*, pp. 4–8, 2006.

[61] G. Papanicolaou, "A new procedure for staining vaginal smears," *Science*, vol. 95, pp. 438–439, 1942.

[62] N. R. Griffin, M. R. Howard, P. Quirke, C. J. O Brien, J. A. Child, and C. C. Bird, "Prognostic indicators in centroblastic-centrocytic lymphoma.," *J Clin Pathol*, vol. 41, no. 8, pp. 866–70, 1988.

[63] K. Belkacem-Boussaid, M. Pennell, G. Lozanski, A. Shana'ah, and M. N. Gurcan, "Effect of pathologist agreement on evaluating a computer-aided assisted system: recognizing centroblast cells in follicular lymphoma cases," in *Proceedings of the IEEE International Conference on Biomedical Imaging: from Nano to Macro*, ISBI'10, (Piscataway, NJ, USA), pp. 1411–1414, IEEE Press, 2010.

[64] N. L. Harris, E. S. Jafe, J. J. Diebold, H. K. Muller-Hermelink, J. J. Vardiman, T. A. Lister, and C. D. Bloomfield, "World health organization classification of neoplastic diseases of the hematopoietic and lymphoid tissues: report of the clinical advisory committee meeting-airlie house, virginia, november 1997.," *Journal of Clinical Oncology*, vol. 17, no. 12, pp. 3835–49, 1999.

[65] A. Schanda, *Colorimetry: Understanding the CIE system.* Wiley-Interscience, New York, 2007.

[66] E. Cosatto, M. Miller, H. P. Graf, and J. Meyer, "Grading nuclear pleomorphism on histological micrographs," in *International Conference on Pattern Recognition*, pp. 1–4, 2008.

[67] J. Kong, H. Shimada, K. Boyer, J. H. Saltz, and M. N. Gurcan, "Image analysis for automated assessment of grade of neuroblastic differentiation.," in *International Symposium on Biomedical Imaging*, pp. 61–64, IEEE, 2007.

[68] T. Cover and J. Thomas, *Elements of Information Theory.* New York: Wiley, 1991.

[69] C. M. Bishop and I. Ulusoy, "Object recognition via local patch labelling," in *Workshop on Machine Learning*, 2005.

[70] P. Combettes, "The foundations of set theoretic estimation," *Proceedings of the IEEE*, vol. 81, pp. 182 –208, Feb. 1993.

[71] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at *http://www.csie.ntu.edu.tw/ cjlin/libsvm*.

[72] J. S. Marron and M. P. Wand, "Exact mean integrated squared error," *Annals of Statistics*, vol. 20, no. 2, pp. 712–736, 1992.

[73] V. C. Raykar, R. Duraiswami, and L. H. Zhao, "Fast computation of kernel estimators," *Journal of Computational and Graphical Statistics*, vol. 19, pp. 205–220, March 2010.

[74] M. P. Wand, "Fast computation of multivariate kernel estimators," *Journal of Computational and Graphical Statistics*, vol. 3, no. 4, pp. 433–445, 1994.

[75] B. W. Silverman, "Algorithm as 176: Kernel density estimation using the fast fourier transform," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 31, pp. pp. 93–99, 1982.

[76] K. Kose, V. Cevher, and A. E. Cetin, "Filtered variation method for denoising and sparse signal processing," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 3329 – 3332, March 2012.

[77] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[78] G. Stockman and L. G. Shapiro, *Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1st ed., 2001.

[79] N. Ray and B. Saha, "Edge sensitive variational image thresholding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '07)*, vol. 6, pp. 37–40, October 2007.

[80] F. Liu, Y. Luo, X. Song, and D. Hu, "Active surface model-based adaptive thresholding algorithm by repulsive external force," *J. Electronic Imaging*, vol. 12, no. 2, pp. 299–306, 2003.

[81] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of International Conference on Computer Vision (ICCV'01)*, vol. 2, pp. 416–423, July 2001.

[82] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr, "Interactive image segmentation using an adaptive gmmrf model," in *Proceedings of European Conference on Computer Vision (ECCV '12)*, pp. 428–441, 2004.

[83] M. Apro, S. Pal, and S. Dedijer, "Evaluation of single and multi-threshold entropy-based algorithms for folded substrate analysis," *Journal of Graphic Engineering and Design*, vol. 2, no. 2, pp. 1–9, 2011.

[84] M.-P. Dubuisson and A. Jain, "A modified hausdorff distance for object matching," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, vol. 1, pp. 566 –568, October 1994.

[85] K. Duman and A. E. Cetin, "Target detection in SAR images using codifference and directional filters," in *Algorithms for Synthetic Aperture Radar Imagery XVII, Proceedings of SPIE*, vol. 7699, April 2010.

[86] M. Dundar, S. Badve, V. Raykar, R. Jain, O. Sertel, and M. Gurcan, "A multiple instance learning approach toward optimal classification of pathology slides," in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR'10)*, pp. 2732 –2735, August 2010.

[87] M. Vilela, J. J. Morgan, and P. A. Lindahl, "Mathematical model of a cell size checkpoint," *PLoS Comput Biol*, vol. 6, p. e1001036, December 2010.

[88] R. Steuer, S. Waldherr, V. Sourjik, and M. Kollmann, "Robust signal processing in living cells," *PLoS Comput Biol*, vol. 7, p. e1002218, November 2011.

[89] L. He, K. J. Isselbacker, J. R. Wands, H. Goodman, C. Shih, and A. Quaroni, "Establishment and characterization of a new human hepatocellular carcinoma cell line.," *J. Cell Physiol.*, vol. 165, no. 3, pp. 459–467, 1985.

[90] B. Bouzahzah, Y. Nishikawa, D. Simon, and B. Carr, "Growth control and gene expression in a new hepatocellular carcinoma cell line, hep40: inhibitory actions of vitamin k.," *In Vitro*, vol. 20, no. 6, pp. 493–504, 1984.

[91] P. Oefinger, D. Bronson, and G. Dreesman, "Induction of hepatitis b surface antigen in human hepatoma-derived cell lines.," *J Gen Virol*, vol. 53, no. 1, 1981.

[92] I. W. Selesnick and K. Y. Li, "Video denoising using 2d and 3d dual-tree complex wavelet transforms," in *Wavelet Appl Signal Image Proc. X, (Proc. SPIE Vol. 5207*, pp. 607–618, 2003.

[93] P. Loo and N. Kingsbury, "Digital watermarking using complex wavelets," in *Proceedings of International Conference on Image Processing (ICIP 00)*, vol. 3, pp. 29 –32, 2000.

[94] G. Chen, T. Bui, and A. Krzyzak, "Palmprint classification using dual-tree complex wavelets," in *Proceedings of International Conference on Image Processing (ICIP 06)*, pp. 2645 –2648, Oct 2006.

[95] M. Thamarai and R. Shanmugalakshmi, "Video coding technique using swarm intelligence in 3-d dual tree complex wavelet transform," in *Proceedings of the Second International Conference on Machine Learning and Computing (ICMLC)*, pp. 174–178, Feb 2010.

[96] A. E. Cetin, O. N. Gerek, and S. Ulukus, "Block wavelet transforms for image coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 3, no. 6, pp. 433–435, 1993.

[97] C. W. Kim, R. Ansari, and A. E. Cetin, "A class of linear-phase regular biorthogonal wavelets," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 4, pp. 673–676 vol.4, 1992.

[98] I. Selesnick, R. Baraniuk, and N. Kingsbury, "The dual-tree complex wavelet transform," *Signal Processing Magazine, IEEE*, vol. 22, pp. 123 – 151, Nov. 2005.

[99] Y. H. Habiboglu, O. Gunay, and A. E. Cetin, "Real-time wildfire detection using correlation descriptors," in *19th European Signal Processing Conference (EUSIPCO 2011), Special Session on Signal Processing for Disaster Management and Prevention*, pp. 894–898, 2011.

[100] K. Duman, "Methods for target detection in SAR images," Master's thesis, Electrical and Electronics Engineering Department, Bilkent University, December 2009.

[101] K. Duman, A. Eryildirim, and A. E. Cetin, "Target detection and classification in sar images using region covariance and co-difference," in *Algorithms for Synthetic Aperture Radar Imagery XVI., Proceedings of the SPIE*, vol. 7337, pp. 73370P–73370P–8, 2009.

[102] A. M. Nazif and M. D. Levine, "Low level image segmentation: An expert system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, pp. 555 –577, Sept. 1984.

[103] S. Chabrier, B. Emile, C. Rosenberger, and H. Laurent, "Unsupervised performance evaluation of image segmentation," *EURASIP Journal of Applied Signal Processing*, vol. 2006, pp. 217–217, January 2006.

[104] R. Maree, P. Geurts, J. Piater, and L. Wehenkel, "Random subwindows for robust image classification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 34 – 40 vol. 1, June 2005.